

MicroPatent® PatSearch Fulltext: Record 1 of 1

Search scope: US Granted US Applications EP-A EP-B WO JP (bibliographic data only) DE-C,B DE-A DE-T DE-U GB-A FR-A

Years: 1981-2005

Patent/Publication No.: ((JP07146820))

[Order This Patent](#)
[Family Lookup](#)
[Find Similar](#)
[Legal Status](#)

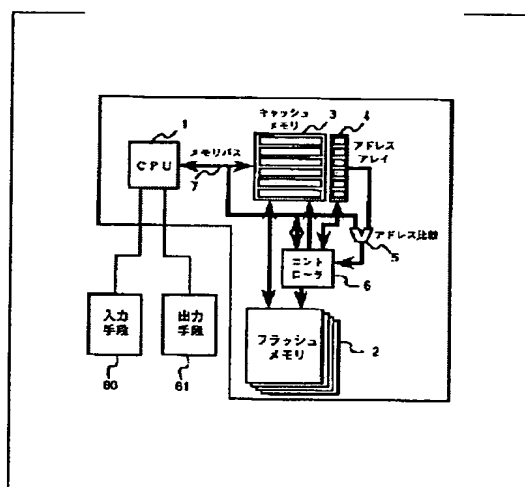
[Go to first matching text](#)

JP07146820 A
CONTROL METHOD FOR FLASH
MEMORY AND INFORMATION
PROCESSOR USING THE SAME
HITACHI LTD

Abstract:

PURPOSE: To perform resume and quick start without providing a backup power source, to inexpensively comprise main memory large in capacity, to simplify system constitution without requiring an auxiliary memory device, to solve a problem of slow rewrite time or impossibility of small amount of random access at that time, etc., and to provide the system constitution in accordance with flash memory of every kind of format by employing the flash memory as the main memory device of an information processor.

CONSTITUTION: The flash memory 2 is connected to the memory bus 7 of a system via cache memory 3 being a nonvolatile memory. An address array 4 which records the information of the address and the access history of stored data, etc., is provided at the cache memory 3. A controller 6 supplies the data in the cache memory 3 or the flash memory 2 to the memory bus 7 by referring to the address to be accessed, or stores the data on the memory bus 7.



[Click here for larger image.](#)

Inventor(s):

KATAYAMA KUNIHIRO
 TSUNEHIRO TAKASHI
 HIDA YASUHIRO
 HATTORI RYUICHI
 KITAHARA JUN
 TOTSUKA TAKASHI
 KAKI KENICHI

BEST AVAILABLE COPY

Application No. 06071049 JP06071049 JP, **Filed** 19940408, **A1 Published** 19950606

Int'l Class: G06F01208

Priority:

JP 05 81642 19930408

JP 05246520 19931001

Patents Citing This One (1):

→ US6791877 B2 20040914 Renesas Technology Corporation
Semiconductor device with non-volatile memory and
random access memory



For further information, please contact:

[Technical Support](#) | [Billing](#) | [Sales](#) | [General Information](#)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-146820

(43) 公開日 平成7年(1995)6月6日

(51) Int.Cl.⁶

G 0 6 F 12/08

識別記号

庁内整理番号

F I

技術表示箇所

3 1 0 Z 7608-5B

審査請求 未請求 請求項の数24 O L (全 27 頁)

(21) 出願番号 特願平6-71049

(22) 出願日 平成6年(1994)4月8日

(31) 優先権主張番号 特願平5-81642

(32) 優先日 平5(1993)4月8日

(33) 優先権主張国 日本 (J P)

(31) 優先権主張番号 特願平5-246520

(32) 優先日 平5(1993)10月1日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 片山 国弘

神奈川県川崎市麻生区王禅寺1099番地株式

会社日立製作所システム開発研究所内

(72) 発明者 常広 隆司

神奈川県川崎市麻生区王禅寺1099番地株式

会社日立製作所システム開発研究所内

(72) 発明者 飛田 庸博

神奈川県川崎市麻生区王禅寺1099番地株式

会社日立製作所システム開発研究所内

(74) 代理人 弁理士 小川 勝男

最終頁に続く

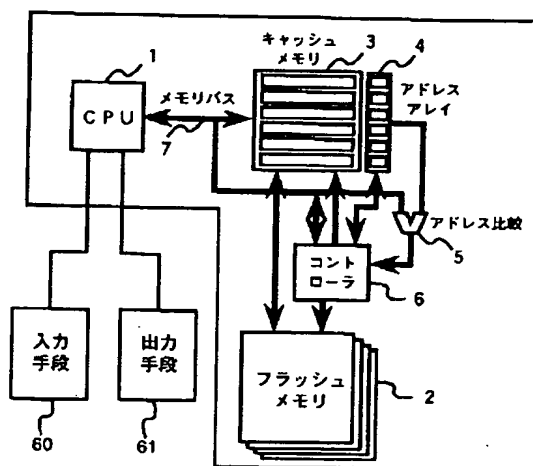
(54) 【発明の名称】 フラッシュメモリの制御方法及び、それを用いた情報処理装置

(57) 【要約】

【目的】 情報処理装置の主記憶装置としてフラッシュメモリを採用することにより、バックアップ電源なしでリジュームやクイックスタートを可能とする。また、大容量の主記憶を安価に構成するとともに、補助記憶装置を不必要としてシステム構成の簡略化を図る。その際、書き換え時間が遅い、小容量のランダムアクセスが不可能などの性能低下原因を解決し、いろいろな形式のフラッシュメモリに対応するシステム構成を提案する。

【構成】 システムのメモリバス7に、揮発性メモリであるキャッシュメモリ3を介して、フラッシュメモリ2を接続する。キャッシュメモリ3には格納しているデータのアドレスやアクセス履歴などの情報を記録するアドレスアレイ4を設ける。コントローラ6が、アクセスするアドレスを参照して、キャッシュメモリ3やフラッシュメモリ2のデータをメモリバス7に供給、あるいはメモリバス7のデータを格納する。

図1



情報処理装置

【特許請求の範囲】

【請求項1】中央処理装置と、データを入力するための入力手段と、データを出力するための出力手段と、データを記憶する揮発性の第一記憶手段と、記憶されたデータを電氣的に書換え可能な記憶手段であって、前記中央処理装置がアクセス可能なアドレス空間を割付けた不揮発性の第二記憶手段と、前記揮発性の第一記憶手段と前記不揮発性の第二記憶手段との相互間でデータのやり取りを少なくとも行うデータ制御手段と、前記揮発性の第一記憶手段に格納されたデータの、前記不揮発性の第二記憶手段におけるアドレスを登録するアドレス登録手段と、該アドレス登録手段に登録されているアドレスと前記中央処理装置がデータアクセスのために出力したアドレスとの比較を行うアドレス比較手段とを具備し、前記データ制御手段は、前記中央処理装置が、データをアクセスするために出力したアドレスを、前記アドレス比較手段に入力し、当該アドレスが、前記アドレス比較手段の出力結果にもとづき、前記アドレス登録手段に登録されている登録アドレスと判断したときには、前記揮発性の第一記憶手段内における登録アドレスをアクセスし、

逆に、前記アドレス登録手段に格納されていないアドレスと判断したときには、少なくとも当該アドレスを含む所定範囲のアドレスを、前記アドレス登録手段に新たに登録し、該新たに登録されたアドレスに対応するデータ格納領域を前記揮発性の第一記憶手段に新たに設定して、前記新たに登録された少なくとも一つのアドレスをアクセスすることを特徴とした情報処理装置。

【請求項2】アクセス履歴記録手段を備え、前記アクセス履歴記録手段は、アクセス履歴情報を記録することを特徴とする請求項1に記載の情報処理装置。

【請求項3】前記データ制御手段は、前記揮発性の第一記憶手段がデータで満たされ、新規にデータを格納する領域が存在しない場合には、前記アクセス履歴記録手段のアクセス履歴情報を参照し、最後に行われたアクセスが最も古いアドレスを検出し、当該アドレスを前記アドレス登録手段から抹消することを特徴とする請求項1または2に記載の情報処理装置。

【請求項4】前記データ制御手段は、前記アドレス登録手段から抹消するアドレスに対応する前記揮発性の第一記憶手段内のデータを、前記不揮発性の第二記憶手段に転送することを特徴とする請求項3に記載の情報処理装置。

【請求項5】前記揮発性の第一記憶手段は、FRAM(Ferroelectric RAM)を有して構成されることを特徴とする請求項1に記載の情報処理装置。

【請求項6】装置に与えられる電源を切断する際に、切断操作により起動する切断手段と、該切断手段による電源の切断処理に対応して、前記中央処理装置が備える内部レジスタの内容および前記入出力手段が備える内部レ

ジスタの内容を含む情報を、前記不揮発性の第二記憶手段の一部の領域に格納する退避処理手段と、該退避処理手段の処理完了後に、装置への電源の供給を遮断する供給電源遮断手段とを備えることを特徴とする請求項1に記載の情報処理装置。

【請求項7】前記データ制御手段は、前記揮発性の第一記憶手段に格納されるデータ量を監視し、あらかじめ定めた容量以上の容量を有するデータ未格納領域を常に確保することを特徴とする請求項1に記載の情報処理装置。

【請求項8】前記不揮発性の第二記憶手段を複数の領域に分割して、各分割領域ごとに、データの書き込みを禁止する旨を表す書き込み禁止フラグを設定可能な書き込み禁止情報格納手段を備え、

前記データ制御手段は、前記不揮発性の第二記憶手段内の、ある領域にデータを格納する際、当該領域に対応して備えられた、書き込み禁止情報格納手段に、前記書き込み禁止フラグが設定されているときには、前記出力手段に、当該領域への書き込み要求がある旨の出力をすることを特徴とする請求項1に記載の情報処理装置。

【請求項9】前記不揮発性の第二記憶手段は、フラッシュメモリを有して構成されることを特徴とする請求項1ないし8に記載の情報処理装置。

【請求項10】前記アドレス登録手段を不揮発性のメモリで構成したことを特徴とする、請求項1に記載の情報処理装置

【請求項11】電源供給手段を備え、電源供給手段が情報処理装置に電源供給を開始する際、前記データ制御手段は、電源供給開始を検知した中央処理装置の指示により、前記アドレス登録手段に保持されているアドレスを参照して、前記不揮発性の第二記憶手段の領域に格納されているデータと、該データが格納されていた揮発性の第一記憶手段内の場所を特定し、前記データを、前記不揮発性の第二記憶手段から読みだし、特定した前記揮発性の第一記憶手段の領域に格納することを特徴とした請求項10に記載の情報処理装置。

【請求項12】電源供給手段を備え、電源供給手段が情報処理装置への電源供給を停止する際、前記データ制御手段は、電源供給停止を検知した中央処理装置の指示により、アドレス登録手段が保持するアドレスを、不揮発性の第二記憶手段の特定の場所に格納し、前記電源供給手段が、再び電源供給を開始する際に、電源供給開始を検知した中央処理装置の指示により、該退避したアドレスを前記アドレス登録手段に戻すことを特徴とする請求項1に記載の情報処理装置。

【請求項13】電源供給手段を備え、電源供給手段が情報処理装置への電源供給を停止する際、前記データ制御手段は、電源供給停止を検知した中央処理装置の指示により、アドレス登録手段が保持するアドレスと、該アドレスに対応する揮発性の第一記憶手段が保持するデータ

を、不揮発性の第二記憶手段の特定の場所に格納し、前記電源供給手段が、再び電源供給を開始する際に、電源供給開始を検知した中央処理装置の指示により、該退避したアドレスを前記アドレス登録手段に戻すことを特徴とする請求項1に記載の情報処理装置。

【請求項14】あらかじめ定めた時間毎に、計測結果を出力する時間計測手段を備え、前記データ制御手段は、時間計測手段が出力する時間計測結果に従い、一定時間ごとに、前記揮発性の第一記憶手段に格納されているデータを、前記不揮発性の第二記憶手段に書き戻すことを特徴とする請求項1に記載の情報処理装置。

【請求項15】あらかじめ定めた時間毎に、計測結果を出力する時間計測手段を備え、前記データ制御手段は、時間計測手段が出力する時間計測結果に従い、一定時間ごとに、前記揮発性の第一記憶手段に格納されているデータと、前記アドレス登録手段に登録されている該データに対応するアドレスを、前記不揮発性の第二記憶手段に書き戻すことを特徴とする請求項1に記載の情報処理装置。

【請求項16】不揮発性の第二記憶手段として、外部とのデータ入出力単位より容量の大きいバッファを内蔵し、該バッファからデータ入出力単位でデータを出力する際は、バッファからの出力データを外部から指定可能で、電氣的に書換え可能な不揮発性メモリを用い、前記データ制御手段は該不揮発性の第二記憶手段から揮発性の第一記憶手段へのデータ転送においては、前記バッファの容量より少ない量を単位とするデータ転送を行い、前記揮発性の第一記憶手段から前記不揮発性の第二記憶手段へのデータ転送においては、前記バッファの容量を単位とするデータ転送を行うことを特徴とする請求項1に記載の情報処理装置。

【請求項17】CPUからデータを書き込む前に、一旦、当該書き込みの対象となるブロック内の全データを消去する必要がある記憶装置であるフラッシュメモリを主記憶装置として用いる情報処理装置において、前記主記憶装置の一部のブロックデータのコピーをそれぞれ保持する複数のデータ領域を有するコピーバック方式のキャッシュメモリと、前記CPUから前記主記憶装置への書き込み処理の際に前記キャッシュメモリがヒットしたとき、前記キャッシュメモリの該当データ領域のデータを更新するとともに、前記主記憶装置の、当該書き込みの対象となったブロックの消去処理を行う制御手段とを備えたことを特徴とするフラッシュメモリを用いた情報処理装置。

【請求項18】前記制御手段は、前記CPUから前記主記憶装置への書き込み処理の際に、前記キャッシュメモリがヒットしても前記主記憶装置内の該当するブロックが既に消去済みの場合は前記消去処理を省略することを特徴とする、請求項17に記載の情報処理装置。

【請求項19】前記キャッシュメモリは、自己がデータ

のコピーを保持している前記主記憶装置内の各ブロックについて、該ブロックが既に消去済みか否かを示す消去情報を保持する手段を有し、前記制御手段は該消去情報を参照し該参照結果に応じて前記消去処理の省略を行うことを特徴とする、請求項18に記載の情報処理装置。

【請求項20】前記制御手段は、前記CPUから前記主記憶装置への書き込み処理の際に、前記キャッシュメモリがミスヒットしたとき、当該書き込みの対象となる主記憶装置のブロックのデータのコピーを保持するために選択されたキャッシュメモリのデータ領域が未更新であれば、そのデータ領域を書き込みデータで更新するとともに、当該書き込みの対象となった前記主記憶装置のブロックのデータの消去を行うことを特徴とする、請求項17に記載の情報処理装置。

【請求項21】前記キャッシュメモリは、各データ領域について、前記CPUからの書き込み処理によってそのデータが更新されたか否かを示す更新情報を保持する手段を有し、前記制御手段は、該更新情報を参照し該参照結果に応じて前記ブロックのデータの消去を行うことを特徴とする、請求項20に記載の情報処理装置。

【請求項22】前記フラッシュメモリの各ブロックのサイズおよび前記キャッシュメモリの各データ領域のサイズがそれぞれmバイト（mは任意の正の整数）であり、前記CPUが前記主記憶装置からmバイトより小さいサイズのデータを読みだす場合、前記主記憶装置内の該当するデータを含むブロックの全データを一旦読みだし、その全データの中から前記CPUが要求する該当データのみをバスへ出力する手段を有することを特徴とする、請求項17に記載の情報処理装置。

【請求項23】前記フラッシュメモリの各ブロックのサイズおよび前記キャッシュメモリの各データ領域のサイズがそれぞれmバイト（mは任意の正の整数）であり、前記CPUが前記主記憶装置からmバイトより小さいサイズのデータを書き込む場合、前記制御手段は、前記主記憶装置内の該当するブロックの全データを一旦読みだし、その全データのうち前記CPUの書き込み処理の対象となる部分領域だけを更新したあと、その1ブロック分の全データを改めて前記主記憶装置内の該当ブロックに書き込むことを特徴とする、請求項17に記載の情報処理装置。

【請求項24】前記キャッシュメモリは、前記データ領域毎に、当該データ領域内のどの部分領域が更新されたかを示す更新領域情報を保持する手段を有し、前記制御手段は、該更新領域情報を参照し該参照結果に応じて前記書き込み処理の対象となる部分領域だけの更新を行うことを特徴とする、請求項23に記載の情報処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、フラッシュメモリなどの電氣的書き込み可能な不揮発性メモリを情報処理装置

の主記憶に用いた場合の制御方法及び、装置に関する。

【0002】

【従来の技術】現在普及しつつある、有望な記憶素子の一つにフラッシュメモリがある。フラッシュメモリを用いることで、不揮発性の主記憶を備えた情報処理装置を構成できる。フラッシュメモリの利点としては、次に示す点が上げられる。

【0003】1. 電源を切ってもデータが保持される (DRAMでは電源を切るとデータが失われる。)

2. SRAMに比べて安価である。

ただし次に示す欠点もある。

1. データを書き込む際は、事前に書き込む領域を消去する必要がある。

2. 消去はチップ単位、または一定サイズのブロック単位で行われる。

3. 1、2の理由により、書き込みに時間を要する。

4. 書き換えにより素子が劣化するので、書換回数に制限がある。

【0004】図17はブロックサイズが512バイト、全メモリサイズが524288バイトのフラッシュメモリの概略を示す図である。図17において110はフラッシュメモリ内の一つのブロック、111はブロック110内の1バイト分のデータ保持部 (本明細書ではこれをセルと呼ぶ) を示す。105は制御回路である。このフラッシュメモリにリードアクセスが発生すると、アドレス信号A9～A18、バッファ131、デコーダ132で決まる該当ブロックの、アドレスA0～A8、バッファ121、デコーダ122で決まる該当セルからデータを読みだし、レジスタ141を介してI/O0～I/O7に出力する。123はバッファ121、デコーダ122の制御信号、133はバッファ131、デコーダ132の制御信号、142はレジスタ141の制御信号である。一方、図17においてライトアクセスが発生すると、アドレス信号A9～A18、バッファ131、デコーダ132で決まる該当ブロックを消去し、該当ブロックの、アドレスA0～A8、バッファ121、デコーダ122で決まる該当セルにI/O0～I/O7からの入力データをレジスタ141を介して書き込む。401は記憶装置104の制御信号である。

【0005】以上述べたフラッシュメモリを記憶媒体とした記憶装置の従来技術としては、例えば、特開平2-292798号公報のフラッシュEEPROMシステムがある。この発明は、フラッシュメモリに欠陥セルが発生した場合の対処方法であり、例えば、代替セルを設けた構成にしたり、欠陥セルの発生により乱れるデータを正規なものに訂正すべく、誤り訂正制御を行うことを提案している。これにより、フラッシュメモリの欠点である書換え回数の制限をカバーし、システム寿命の長期化を図る効果を得ている。また、システムにライトキャッシュメモリを搭載して、キャッシュメモリへの最後の書

き込みからの経過時間に基づいて、フラッシュメモリへのライトバックを行なう。頻繁に書き変わるデータを、フラッシュメモリ上で書換えずに、キャッシュメモリ上で書換えることにより、フラッシュメモリの動作を少なくし、システムの寿命の長期化を図ることを目的としている。

【0006】一方、高性能なパーソナルコンピュータなどでは、リードまたはライト時間を高速化する手段として、DRAM-SRAMキャッシュシステムがよく用いられる。一般にキャッシュメモリは、CPUとアクセスに時間を要する記憶装置の間に設けられてバッファメモリの機能を果たし、CPUが記憶装置をリードした場合は、そのアドレス、データ等をキャッシュメモリに格納しておき、次にCPUが記憶装置の同じアドレスをリードした時はそのアドレスに対応するデータをキャッシュメモリから得ることによりアクセスを高速化する。キャッシュメモリにはライトスルー方式とコピーバック方式の2方式が知られている。ライトスルー方式は、記憶装置への書き込み要求に対してキャッシュメモリとともに記憶装置も同時に書き換える方式である。これに対してコピーバック方式は、記憶装置への書き込み要求に対して、キャッシュメモリのみ書き換えを行ない、処理に時間を要する記憶装置の書き換えを省略することによりアクセスを高速化する方式である。

【0007】また、フラッシュメモリのチップコストを低減するために、フラッシュメモリのチップ面積を小さくするセル構造が注目されている。たとえば、いわゆるNAND構造を有したフラッシュメモリがこれに相当する。この構造では、ランダムアクセスではなく、ラインアクセスを行い、データの入出力方法はライン単位でシリアルに行う。今後のフラッシュメモリは、メモリの高集積化を目指したラインアクセス方式が主流になると考えられる。他にも、前記ラインアクセスに類似したアクセス方式を採用するメモリの開発が進んでいる。

【0008】

【発明が解決しようとする課題】特開平2-292798号公報記載の従来技術は磁気ディスク装置置き換えを主目的としていると考えられる。外部とデータをやり取りするためにシステムに備えられている外部I/Oバスによる、セクタ単位でのアクセスを前提としている。しかし、フラッシュを主記憶に採用した場合のCPUからのランダムアクセス、すなわち数バイト等の少ない単位での直接のデータのやり取りについては全く考慮されていない。上述の代替セル方法や、誤り訂正符号処理は、データのやり取りをセクタ単位で行うことを考慮している。バイトあるいはワード単位でのやり取りは不可能である。また、あるセクタ内で不良セルが発生すると、対応した適切な処理を行うための時間が必要となる。その結果、データの転送時間が長くなり、データの転送が遅れてしまう事態を招く。特に誤り訂正符号を使用した処

理は、複雑なものとなる。また、同発明におけるキャッシュ方式は、システム寿命の長期化のために設けられたものである。データ書き込み時のキャッシュデータの操作についての技術開示はなされているが、外部システムであるホスト側へのリードデータの転送については、全く技術開示がされていない。したがって、高速アクセス可能な手段の提供とはいえない。

【0009】現在、パーソナルコンピュータ等の情報処理装置において、一般的に使用されているキャッシュメモリシステムは、通常、DRAM（ダイナミックラム）を有して主記憶を構成し、さらに、SRAM（スタティックラム）を有して構成したキャッシュメモリを備え、CPUの動作速度に、DRAMのアクセス動作が追いつけないことに対処している。つまり、アクセスされたアドレスを、SRAMに割付け、アクセス速度の遅いDRAMは、データのバックアップ用に使用して、CPUからは、アクセス速度の速いSRAMが、主記憶となっているように見せかける技術である。この技術においては、DRAMのアクセススピードとSRAMのアクセススピードでは、数倍程度の違いがあるが、違いが大きな場合でも、10倍以内であるため、キャッシュメモリに割当てされていないアドレスに対するライトアクセスの場合、すなわち、ライトミスヒット時のリカバリタイムは、それほど大きなものではない。フラッシュメモリを主記憶に採用した場合には、フラッシュメモリの書換え時間は、DRAMと比較すると数桁の違いがあり、ライトミスヒット時のリカバリタイムは、非常に大きくなってしまい、システムの性能の低下を招くことになる。したがって、この点を考慮したシステムを実現する必要がある。

【0010】また、上述のラインアクセス方式のメモリを用いて、ランダムアクセスを高速に行うことが今後重要な技術となる。前記特開平2-292798号公報を含め、従来の技術においては、この点を考慮したものがない。また、主記憶が不揮発性になることで、システム上、大きな違いが生じる。例えば、主記憶が揮発性であれば、補助記憶装置を備えて、ファイルデータの保存を行うのが一般的である。しかし、不揮発性の主記憶システムの場合は、補助記憶装置を備える必要が無く、主記憶がデータ保存の領域となる。このことが、不揮発性の主記憶を備えるシステムの効果のひとつであるが、逆に、このために発生する問題もある。例えば、各種の情報処理装置では、プログラムや操作のミスにより装置が暴走してしまうという事故が起こることがある。この際、揮発性の主記憶であれば、最終手段として、ハードリセットをかけたり、電源を一度切断して再スタートすることが可能である。これによって主記憶の内容をクリアし、補助記憶装置から再度、データを主記憶にロードして、正常な状態に戻すことができる。しかし、不揮発性の主記憶では、装置の暴走により、主記憶内に存在す

るデータが破壊されると、正しいデータが存在しなくなり、正常な状態に戻すことが困難となる。

【0011】また、キャッシュメモリだけが書き換えられている状態で、システムの電源が切断された場合、主記憶が揮発性のシステムでは、主記憶およびキャッシュメモリのいずれの記憶データも揮発するため問題は発生しない。しかし、主記憶が不揮発性の場合には、キャッシュメモリだけに最新のデータが格納された状態で、電源が落される事態が生じると、ユーザがデータが残されているつもりでも、実際には、ユーザが入力したつもり

のデータが、キャッシュメモリから揮発し、消滅してしまうことが起こりうる。したがって、不揮発性の主記憶を備える情報処理装置では、装置の暴走や、電源遮断に対処する方式の提供が必要となる。また、上述のキャッシュメモリが揮発性の場合、電源遮断後の再起動時に、初めのうちはミスヒットが続き、アクセス速度が劣化するという問題が生じる。

【0012】本発明の目的は、フラッシュメモリを主記憶に備える情報処理装置を実現することである。具体的には、CPUからのランダムアクセスに対応可能とすることである。さらには、高集積なラインアクセス方式のフラッシュメモリに対応できる情報処理装置を実現することである。さらには、その際のミスヒットアクセスに対するリカバリタイムを短くして性能を向上させることである。さらには、プログラムや操作のミスによる暴走による重要なデータの破壊を防止することである。さらには、キャッシュメモリ内の情報と、主記憶内の情報が一致していない状況での電源が遮断された時の対処を考慮して、信頼性を向上させることである。さらには、動作の停止後、再開時のアクセス性能を向上させ、総合的な使用感の向上を目指すことである。

【0013】

【課題を解決するための手段】上記目的を達成するために、本発明の情報処理装置は、中央処理装置と、データを入力するための入力手段と、データを出力するための出力手段と、データを記憶する揮発性の第一記憶手段と、記憶されたデータを電氣的に書換え可能な記憶手段であって、前記中央処理装置がアクセス可能なアドレス空間を割付けた不揮発性の第二記憶手段と、前記揮発性の第一記憶手段と前記不揮発性の第二記憶手段との間でデータのやり取りを少なくとも行うデータ制御手段と、前記揮発性の第一記憶手段に格納されたデータの、前記不揮発性の第二記憶手段におけるアドレスを登録するアドレス登録手段と、該アドレス登録手段に登録されているアドレスと前記中央処理装置がデータアクセスのために出力したアドレスとの比較を行うアドレス比較手段とを備えたことを特徴とする。

【0014】そして、前記データ制御手段は、前記中央処理装置が、データをアクセスするために出力したアドレスを、前記アドレス比較手段に入力し、当該アドレス

9

が、前記アドレス比較手段の出力結果にもとづき、前記アドレス登録手段に登録されている登録アドレスと判断したときには、前記揮発性の第一記憶手段内における登録アドレスをアクセスし、逆に、前記アドレス登録手段に格納されていないアドレスと判断したときには、少なくとも当該アドレスを含む所定範囲のアドレスを、前記アドレス登録手段に新たに登録し、該新たに登録されたアドレスに対応するデータ格納領域を前記揮発性の第一記憶手段に新たに設定して、前記新たに登録された少なくとも一つのアドレスをアクセスするように構成する。

【0015】さらに、本発明の情報処理装置は、CPUからデータを書き込む前に、一旦、当該書き込みの対象となるブロック内の全データを消去する必要がある記憶装置であるフラッシュメモリを主記憶装置として用いる情報処理装置において、前記主記憶装置の一部のブロックデータのコピーをそれぞれ保持する複数のデータ領域を有するコピーバック方式のキャッシュメモリと、前記CPUから前記主記憶装置への書き込み処理の際に前記

【0016】

【作用】上記手段は、具体的には、例えば以下のように動作する。外部I/Oバスを通さずに、直接CPUがフラッシュメモリの格納データをアクセスできるようにするためには、高速ランダムアクセスを実現しなければならない。フラッシュメモリを直接アクセスすると、ランダムアクセス型のフラッシュメモリでは、リードアクセスはDRAMなみの高速ランダムアクセスが可能であるが、ライトアクセスでは、書換えがブロック単位でしかできず、しかも非常に遅い。そして先述のように、今後主流になると考えられる、ラインアクセス方式のフラッシュメモリでは、リードアクセスにおいても、ランダムアクセスには適さない。そこで、キャッシュメモリとして動作するバッファメモリを、フラッシュメモリとCPUの間に設けた構成とする。このキャッシュメモリは、高速ランダムアクセスが可能な、DRAMやSRAMで構成するのが最適である。

【0017】そして、このキャッシュメモリに対応して、格納してあるデータのアドレスを記録するアドレスアレイと、アクセス履歴を記録する記憶手段を備える。そして、フラッシュメモリの書換えが、キャッシュメモリのアクセスに対して非常に遅いことに対処するため、キャッシュメモリには、未格納のアドレスのデータを常に書き込めるように、空き領域をあらかじめ用意しておく。また、装置の暴走による、主記憶データの破壊に対処するため、各データ領域に、ライトプロテクトフラグを備えた構成とし、このフラグにより、書換え禁止状態のデータの書換えを行った場合には、CPUに割込みを

10

かけて警告を行う機能を設ける。また、フラグを書き変える際にも、割込みによる警告を行うことにより、2重の安全が図れる。そしてその後、特定の書き込み要求レジスタにCPUが所定コードを書き込まなければデータの書換えを許可しないようにしておく。

【0018】さらに高速化を図るために、キャッシュメモリ上のデータがCPUにより書換えられたら、そのデータに対応するフラッシュメモリ上の領域をあらかじめ消去し、該データの書き戻しの際に、フラッシュメモリの当該領域を消去するための時間を節約できるようにする。また、前記アドレス登録手段のデータを、装置への電源供給を遮断しても、揮発させることなく、再び電源供給を開始して、装置の動作を再開した際に、保持されているようにし、このデータをもとに、キャッシュメモリ内のデータを復元する。さらに具体的に説明する。キャッシュメモリを設けることにより、書換えが遅いフラッシュメモリの欠点をカバーできるとともに、シリアルバッファによって、外部インタフェースが接続されており、高速なランダムアクセスが不可能なフラッシュメモリに対しても、キャッシュメモリのランダムアクセスは可能である。これらは、見かけ上、CPUのメモリバスに直結した、フラッシュメモリを備える主記憶として、使用することが可能となる。

【0019】上述のように、キャッシュメモリに格納しているデータの論理アドレスを記録するアドレスアレイと、キャッシュメモリに格納しているデータの古さを示すアクセス履歴を記録する記憶領域を設けることにより、まず、キャッシュメモリに存在するアドレスのアクセスであるか否かを判断できる。次に、キャッシュメモリに存在しないアドレスのアクセスであれば、アクセス履歴上、最後のアクセスが起きてから、最もアクセスされていないデータを探し出して、主記憶たるフラッシュメモリにライトバックし、空き領域を作りだして、新たなデータを格納するという動作を行うようにする。これは、キャッシュメモリのリプレースメントアルゴリズムとして、知られているものである。

【0020】また、キャッシュメモリ内に、常に空き領域を確保しておくことにより、CPUからデータの書き込み要求があり、そのデータのアドレスがキャッシュメモリに格納されていない場合に、要求があってから、該データを格納するための空き領域を作り出していたのでは、書き込み動作の遅いフラッシュメモリでは大きな性能劣化となるため、あらかじめ用意してある空き領域に、とりあえずデータを格納し、CPUからの書き込みデータ転送が終了してから、フラッシュメモリ内に空き領域を作り出す処理を始めればよい。

【0021】さらにまた、装置の暴走に対処するためのライトプロテクトフラグを備えた構成とし、このフラグの書換えや、書き込み禁止領域の書換え時に、CPUに割込みを行い、これに対応してCPUが警告を行う構成

にしておくことによって、CPUが、異常な書換えを行おうとしているか否かが把握可能となる。そして、CPUが、特定のレジスタに対しての書換えを行わなければ、格納データの書換えを行わないようにすることにより、装置暴走等の異常時の主記憶内のデータの書換えを防ぐことができる。なお、CPUでは、上記の割込みを受けとった際には、ユーザに対して、データの書換えを行うべきか否かを確認するルーチンを実行する構成としておけばよい。ユーザは、この確認ルーチンで、装置が暴走しているか否かを判断して、適宜、情報処理装置に指示を与える事になる。この場合、もちろん、装置の動作の中断や再開等の指示が可能な構成にしておくのが好ましい。

【0022】また、図18を参照し、本発明による情報処理装置の作用を説明する。CPU101から記憶装置104へのリードアクセスでキャッシュメモリブロック300がヒットした場合は、キャッシュメモリブロック300から目的のデータを読み出す。リードアクセスでキャッシュメモリブロック300がミスヒットした場合は、CPU101は記憶装置104から直接データをリードする。これに伴い、そのデータとアドレスはキャッシュメモリブロック300内の新たに選択されたレジスタに保持される。新たなレジスタの選択は、先述の置き換え操作（リプレースメントアルゴリズム）にしたがって行われる。尚、この新たに選択されたレジスタ内に前回以前のアクセスで更新されたアドレスとデータが保持されている場合は、そのアドレスとデータを記憶装置104へ書き戻した後、新たな書き込みアドレスとデータを保持する。本発明では記憶装置104はフラッシュメモリで構成されるので、その書き込みの際に、該当ブロックが未消去の場合はその該当ブロックを消去してから書き込みを行う。

【0023】一方、本発明のキャッシュメモリブロック300はコピーバック方式のキャッシュなので、CPU101からのライトアクセスでキャッシュメモリブロック300がヒットした場合は、キャッシュメモリブロック300内の該当レジスタを更新するだけで、記憶装置104への書き込みは省略される。すなわち、後にリプレースされる際に記憶装置104の該当するブロックへ書き戻される。換言すれば、この記憶装置104の該当ブロックのデータはいずれ消去される。したがって、本発明では、そのブロックがまだ消去されていないときには、そのブロックの消去を前もって行う。この事前消去処理では、CPU101はその消去が完了したブロックに直ちに書き込みを行うわけではないので、その消去処理の完了を待つ必要はない。以後、そのブロックへのデータの書き戻しを行う必要が発生した際には、それに先立つ消去処理が省略される（既に実行されている）のでアクセス時間が短縮される。

【0024】CPU101からのライトアクセスでキャ

ッシュメモリブロック300がミスヒットした場合は、キャッシュメモリブロック300内のリプレース対象レジスタが新たに選択され、このレジスタが更新済みであれば、このレジスタに格納されている更新済みのデータを記憶装置104へ書き戻してから、CPU101からの当該アクセスアドレスとデータをキャッシュメモリブロック300内の該当レジスタに格納する。更新済みでなければ、リプレース対象レジスタを更新する。上述した記憶装置104へ書き戻す処理は、キャッシュメモリブロック300のリードアクセスミスヒットの際の記憶装置104への書き込み処理と同様である。

【0025】次に、アドレス登録手段のデータを保持することについての動作を説明する。キャッシュメモリに格納されているデータは、CPUから高速にリードライトできるようにするために、一時的にキャッシュメモリに置かれているデータである。そしてキャッシュメモリの容量は限られているため、最もCPUからアクセスされる可能性と頻度が高いとされるデータが選択されて置かれている。もしキャッシュメモリ上に置かれていないデータが連続してアクセスされると、著しく性能を劣化することになる。この事態が起きることがないように、キャッシュメモリの構成や、リプレースメントアルゴリズムなどを最適化する必要がある。しかし如何に最適化しても、キャッシュメモリのデータが揮発してしまい、キャッシュメモリに全くデータが格納されていない時には、全てのリードアクセスはミスヒットとなり、性能を著しく劣化することになる。キャッシュメモリがある程度のデータを主記憶から取り出して来るまでは、リードアクセス速度は遅くなる。しかし、電源供給停止によって、キャッシュメモリのデータが揮発してしまっても、アドレス登録手段の比較的少ないデータさえ保持されていれば、キャッシュメモリ内のデータを、電源供給遮断前の状態に復帰させることが可能である。つまりアドレス登録手段に登録されているアドレスのデータを、主記憶からキャッシュメモリに展開すればよい。アドレス登録手段のデータ保持の方法としては、一つは、アドレス登録手段のメモリを電池バックアップでデータを保持する。または、アドレス登録手段をランダムにリードライトが可能な不揮発性のメモリを使用する。そして他の方法として、電源供給遮断時に、不揮発性メモリである主記憶メモリの一部、あるいは同等のメモリに、アドレス登録手段のデータを退避し、電源供給再開時に再びアドレス登録手段に書き戻すという手段がある。

【0026】

【実施例】以下、本発明の実施例を図面を参照して説明する。図1は、本発明の第1の実施例の構成図である。図1において、1は、プログラムの実行や、データの処理を行うCPU（中央処理装置）、2は、CPUが処理実行するためのプログラム、データ等を格納する大容量の不揮発性メモリであるフラッシュメモリ、3は、フ

ラッシュメモリ2に格納されていて転送されてきたデータやCPU1からの書き込みデータ等を、一時的に記憶する揮発性メモリであるキャッシュメモリである。このキャッシュメモリは、例えば、DRAM（ダイナミックラム）、SRAM（スタティックラム）等によって実現可能である。4は、キャッシュメモリ3が格納しているデータに割り付けられているCPUアドレス（CPUがデータのアクセスのために出力するアドレスをいう）と、それに付随した情報を記録するアドレスアレイ、5は、CPU1がアクセスを要求したデータに対応するアドレスと、アドレスアレイ4に記録されているアドレスを比較するアドレス比較回路、6は、CPU1のアクセス要求に、可能な限り高速に、正確なデータをアクセスできるように統一管理制御するコントローラ、7はCPU1のメモリバスである。アドレスアレイ4、アドレス比較回路5、コントローラ6は、例えば、各種CMOS、トランジスタ、抵抗、ROM（所定の処理を行うプログラムが格納される）、RAM、CPU等の電子デバイスにて実現可能である。なお、CPUがアクセスするアドレスが存在するアドレス空間は、例えば、フラッシュメモリ2に対して割り当てられているとして説明を行う。60は、データをアクセスするためのコマンド等の各種コマンドや、アドレス等を少なくとも入力する機能を有する入力手段である。例えば、マウス、キーボード等によって実現される。61は、後述するメッセージや、必要な情報を出力する機能を有する手段であり、例えば、プリンター等の印刷手段や、CRT、ELディスプレイ、液晶ディスプレイ等の表示手段によって実現できる。

【0027】次に、本構成における動作について説明する。CPUと記憶装置を備える、一般的な情報処理装置では、主記憶装置として、DRAM等の揮発性メモリを使用して構成するのが一般的であるが、これは、アクセス速度を考慮してのことである。これに対して、例えば、携帯型パソコン等でニーズの高い、いわゆるリジューム機能、クイックスタート機能等の実現には、不揮発性メモリを使用することが必要であるが、バッテリーでバックアップを行うSRAMでは、容量的にも、信頼性上も問題点が多く、また、電氣的書換え可能なROMでは、書換え時間が非常に遅いため、これを使用することは難しい。そこで、本実施例では、揮発性メモリをデータ等の一時的な記憶を行うキャッシュメモリとして採用し、CPU1が備えるメモリバス7に、直接接続して、実質的な主記憶装置として、フラッシュメモリを使用した構成とする。つまり、主記憶装置であるフラッシュメモリ2は、CPU1から直接アクセスされることはなく、したがって、主記憶装置のアクセスの遅さが、CPU1の動作に殆ど影響を及ぼさないようにするものである。

【0028】図1を参照して説明を続ける。CPU1

は、メモリバス7を介して、キャッシュメモリ3をアクセスする。ただし、この時アクセスするアドレスは、アドレス比較回路5に入力され、アドレスアレイ4に予め登録されているアドレスと比較する。もし、登録されているアドレスであれば（以下、これを「アドレスヒット」と称する）コントローラ6はキャッシュメモリ3内における当該アドレスに相当する個所をアクセスする。逆に、アドレスアレイ4に登録されていない（以下、これを「アドレスミスヒット」と称する）アドレスであれば、アドレスアレイ4に、当該アドレスを登録する。その後、当該アドレスに対応するデータをキャッシュメモリに転送、格納し、当該アドレスに対応するフラッシュメモリ2内の領域のアクセスを行う。

【0029】図2に、かかるコントローラ6の構成図を示す。図2中、11は、高速に連続アドレスを発生して、メモリを高速にアクセスするためのDMAコントローラであり、数十〜数百バイトの連続データの転送時に使用する。12は、メモリ制御信号発生回路で、フラッシュメモリを始め、キャッシュメモリ、アドレスアレイ等の動作を制御するための制御信号を生成する。例えば、CPU、ROM、RAM、各種ゲート、ROMに内蔵されるプログラムによって実現できる。なお、キャッシュメモリ等でDRAMを使用している場合は、DRAMをリフレッシュするための、リフレッシュコントローラも、前記メモリ制御信号発生回路が備える構成にすれば良い。13は、メモリ制御タイマであり、フラッシュメモリの内容の消去、書き込み等の時間を測定する手段であり、CPU、ROM、RAM、各種CMOS、ROMに内蔵されるプログラムによって実現できる。14は、揮発性のデータメモリであり、制御プログラム実行のワークエリアとして、また、メモリバスからのデータを一時的に蓄積したり、アドレスアレイの情報や、システム内で授受されるデータを一時的に保存する領域として使用される。15は、制御プログラムが格納されるROM、16は、制御プログラムを実行し、本発明のかかるシステム全体を統括して制御するプロセッサ、17は、システム内でやりとりされるアドレスや、システム内でやりとりされるデータ等を伝送するためのバスである。

【0030】このコントローラ6の動作、すなわち前記制御プログラムが行う処理を説明するためのフローチャートを図3に示す。図3に示すように、まず、CPU1から、あるアドレスに対してのアクセス要求が出される（ステップa）。このアドレスはアドレス比較回路5に入力され、アドレスアレイ4に登録されているアドレスと比較処理される（ステップb）。アドレスヒットであれば、所望のデータがキャッシュメモリ3内に存在することになる。そこで、このアドレスをアドレスアレイ4に記憶されている情報にもとづいてキャッシュメモリ3の該当するアドレスに変換処理し（ステップc）、この

変換処理されたアドレスに従って、キャッシュメモリ3の該当領域をアクセスする(ステップd)。すなわち、リードアクセスであれば、キャッシュメモリ3内のデータをメモリバス7を介してCPU1へ出力し、ライトアクセスであれば、キャッシュメモリ3の該当する場所を、メモリバス7を介してCPU1から入力されるデータに書換える。一方、アドレスミスヒットであれば、キャッシュメモリ3内に、新規に、データを格納するためのデータ領域を作り出す(ステップe)。なお、ステップeにおける処理の詳細については、本フローチャートの説明の後で述べることにする。

【0031】さて、新規のデータ格納領域が作られたならば、当該領域に対して、アクセスを行う。もしリードアクセスであれば、フラッシュメモリ2から該当するアドレスに格納されているデータを読み出して、キャッシュメモリ3に転送、格納し(ステップf)、さらに、メモリバス7に当該データを出力する(ステップg)。もし、ライトアクセスであれば、CPU1からのデータをキャッシュメモリ3の新規のデータ格納領域に書き込む(ステップh)。そして、最後に、アドレスアレイを新規書き込みに合わせて更新する。すなわち、ミスヒットであるアドレスをアドレスアレイに登録する(ステップi)。以上が、コントローラ6の動作の概略である。さて、前述した、ステップeの処理について説明する。なお、コントローラ6は、揮発性記憶手段に格納されるデータ量を監視し、該データ量が予め定めた値をこえないように、予め定めた容量以上のデータ未格納領域を設ける処理を行うように構成しておくのが好ましい。

【0032】仮に、この情報処理装置が動作開始後、それほど処理を行っていない段階であれば、キャッシュメモリ3内には、データの未使用領域が多数存在しているので、特別の処理をしなくとも、この未使用領域内に、新規格納領域を割り当てればよい。しかしながら、未使用領域が存在しなくなれば、既に格納しているデータを消去してでもデータ格納領域を作り出さなければならない。そのために、アドレスアレイ4に登録されている各アドレスに対応して、CPUがどのような順序でアクセスしたかを示す情報である「アクセス履歴」を記憶しておく。キャッシュメモリ3に格納されているデータのうち、現時点以降、最もアクセスされないと予想されるものを探し出す。これが、キャッシュメモリ3内にアクセス対象となるデータが存在する割合(以下「ヒット率」と称する)と、アクセス速度の高速化等の性能を向上させる。

【0033】そのためには、各アドレスが最後にアクセスされた相対的な古さを示す情報が記録されていればよい。つまり、この記録内容にもとづいて、前回のアクセスが行われたのが、最も古い(時間軸上で最も過去方向に存在する)アドレスを探し出すことができ、キャッシュメモリ3内からデータを消去しても、最も問題のない

ものと思われるものを選定できる。このアクセス履歴に関する情報は、アドレスアレイ4に格納すると、処理速度の向上等の点から効率的である。ただし、アクセス履歴は、キャッシュメモリ容量が大きくなると、相対的な古さを記録するためには、非常に複雑、かつ、大容量になるため、工夫が必要である。

【0034】その工夫として、ヒット率が大きく減少してしまうが、アクセスの来歴記録を簡略化することが可能である。例えば、消去するデータをランダムに選択する方法が考えられる。あるいは、キャッシュメモリ3の先頭から順にデータを書き込むことにしておき、データの消去も、先頭から順番に行えば、古く書き込まれた順にしたがってデータを消去する、いわゆるFIFO制御が簡単に実現可能である。または、一般的なキャッシュメモリシステムで使用されているセットアソシエティブ方式を採用すれば、アドレス比較も単純化する。

【0035】そこで、このセットアソシエティブ方式について図4を参照して説明する。今、n個の連続したアドレスで構成される領域を、一つのブロックとして考え、キャッシュメモリは、数ブロックのデータで構成されているものとする。図4は、この方式における構成例を示している。21は、アドレスアレイ、22は、キャッシュメモリ3に格納されているデータに1対1で対応する、1つのアドレスを格納するアドレスアレイ21内のアドレス格納領域、23は、アドレス格納領域22を複数個有して構成されるブロックである。図4に示す実施例では、ブロック23が4つずつ存在する。これが、4ウェイセットアソシエティブ方式の構成である。24は、CPUがメモリデータをアクセスするために出力したCPUアドレス、25、26は、それぞれ、CPUアドレス24を、セットアソシエティブ方式に適用するために分割したもので、25を「ブロックアドレス」と称し、26を「セットアドレス」と称する。

【0036】セットアドレス26は、CPUアドレス24のうち下位側のアドレスであり、そのビット数は、ブロック23を構成するアドレス格納領域22の個数で決まる。例えば、ブロック23が256個のアドレス格納領域22を有して構成されておれば、セットアドレスは、8ビットとなる。一方、ブロックアドレス25は、CPUアドレス24の上位側のアドレスであり、そのビット数はCPUアドレス24のビット数から、セットアドレス26のビット数を引いた数となる。ただし、CPUアドレス24によりアクセスするメモリデータのバイト数は、キャッシュメモリとフラッシュメモリ間のデータ転送時のバイト数により決まる。例えば、これが16バイトであるとすれば、CPUアドレスの最下位3ビットについては、アドレス比較する必要はない。そのため、セットアドレスはCPUが実際に出力するアドレスのうち、これに相当するビットを除くことができる。27は、セットアドレス26により選択されてアクセスさ

れた、キャッシュメモリに登録されているブロックアドレスを示す。個々でアドレスアレイ21により出力されたアドレス値に対応するデータは、キャッシュメモリに格納されていることになる。4ウェイセットアソシエティブ方式では、一つのセットアドレスに対して4つのブロックアドレスが存在する。28は、CPUアドレスのブロックアドレス25と、アドレスバッファ21より出力されたブロックアドレス27が一致するか否かを検出するアドレス比較回路であり、4ウェイセットアソシエティブ方式であることから、4つの比較回路28を設けている。この比較回路は、例えば、各種ゲート等にて実現できる。

【0037】次に、具体的に数値を設定して説明する。主メモリ容量を16MBとし、CPUアドレスが24ビットとする。また、フラッシュメモリとキャッシュメモリの制御方式として、前述の4ウェイセットアソシエティブ方式を採用し、1ブロック内の単位データ(16バイト単位)格納領域数を256とする。

【0038】従ってキャッシュメモリ容量は
 $16 \text{ バイト} \times 256 \times 4 \text{ ウェイ} = 16 \text{ KB}$

と算出できる。まず、CPUが主メモリデータをアクセスするために、24ビットのCPUアドレスを出力する。アドレスヒット、ミスヒットの判定では、CPUアドレスの下位4ビットは必要ないので、上位側20ビットによりその判定を行う。このうち、セットアドレス26は256の格納領域が存在するため、8ビットであることになる。つまり、20ビットのアドレスのうち下位から8ビットを使用して、アドレスアレイ21をアクセスすることになる。

【0039】この時、ブロック数が4つ存在するため、4か所をアクセスして、4つのブロックアドレス27を得る。尚、ブロックアドレスは、20ビットから8ビットを引いた12ビットのデータである。このデータとCPUアドレス24のブロックアドレス25を、比較回路28を使用して一致するか否かの検出を行う。4つの比較回路が、全て不一致状態を検出すれば、アドレスミスヒットであり、キャッシュメモリ内に所望のデータは存在しないため、フラッシュメモリをアクセスすることになる。もし、一つでも一致状態にあることを検出すれば、アドレスヒットであり、キャッシュメモリ内の該当する個所をアクセスする。以上により、4ウェイセットアソシエティブ方式の動作が完了する。なお、回路を簡略化するために、ブロック数を減らせばヒット率が減少することになるし、ブロック数を増加すればヒット率は増加するが、比較回路やアドレスアレイが複雑化することになる。ブロック分けをせずに、全データアドレスアレイの内容と、CPUアドレスを比較するための比較回路を設けた構成にすることも、もちろん可能ではある。しかしながら、プログラム実行においてはアクセスするアドレスは局所性をもつため、ある程度ブロック分けし

た方式を採用した方が、ハードウェア構成上効率的である。あるいは、アドレス比較回路の行う機能を、ソフトウェアで実現することも容易であるが、その際には、アドレス比較器の数が、ソフトウェアにおけるアドレス比較処理の回数になるため、処理速度の著しい低下などを招く可能性がある。

【0040】次に、4ウェイセットアソシエティブ方式を採用した場合のアクセス履歴の記録方法について説明する。4ウェイセットアソシエティブ方式においては、下位アドレスが連続している一連のデータを4通り格納することになる。したがって、同一の下位アドレスのデータが最高4つ存在することになり、その4つに対して、アクセス履歴が比較できれば、必要かつ十分である。そこでアクセスが行われるたびに、アクセスされたグループのアクセス履歴に、関する情報を更新し、アクセスが行われたデータの履歴情報を「最新」として、それ以外のデータの新鮮さを1段階ずつ下げていく。

【0041】例えばA、B、C、Dの4つのデータのアクセス履歴が「3」、「4」、「1」、「2」の新鮮さの順序(数が小さいほど、新鮮さの度合いが高いことを意味する)を有していたとする。次にAがアクセスされた場合、アクセス履歴は「1」、「4」、「2」、「3」となる。つまり、Bはもともと最も古いデータであるため、アクセス履歴はそのままであり、C、Dのアクセス履歴が、それぞれ繰り下がることになる。そして、新規にデータの格納領域を作り出す場合には、アクセス履歴が4番目になっているデータが、最もアクセスが起きていないデータと判断されるため、フラッシュメモリに書き戻せば(ライトバック)よい。もし、このようなセットアソシエティブ方式を使用せずに、全てのデータの相対的なアクセス履歴を記録していたら、膨大な処理が必要になる。したがって、セットアソシエティブのセット数も、システム規模と処理効率のトレードオフを考慮して決定するのが好ましい。なお、以上の処理に必要な情報を、アドレスアレイに全て格納しておけば良い。

【0042】ところで、これまで説明した動作では、CPU1からあるアドレスへの書き込み要求が発生し、それがアドレスミスヒットであった場合に、実用上若干の問題がある。というのは、キャッシュメモリ内の、あるアドレスデータをフラッシュメモリにライトバックして、新規にデータの格納領域を作ることになるが、フラッシュメモリのデータ書換えには、非常に時間がかかるため、処理速度の低下などのシステム性能の大きな低下を招いてしまう。そこで、コントローラ6の制御を、図5に示すようにする方が、システム性能の低下を防ぐことができ、実用的である。図5は、図3と殆ど同じ処理を示すため詳細な説明は省くが、ステップeにおける新規格納領域の作成タイミングが、図3に示す処理フローとは異なる。CPU1からのデータ転送が終了した時点

で、新規格納領域の作成動作に入る。つまり、常に新規格納領域を用意しておいて、CPU1からの書き込み要求がアドレスミスヒットであった場合には、その新規格納領域に書き込む。これにより、フラッシュメモリの書換え時間が、CPU1の動作に影響を与えないようにする。ただし、この方法では、キャッシュメモリ3にデータの存在しない領域を常に用意しておく分、キャッシュメモリの格納データ容量が少なくなるため、ヒット率は減少することになることを考慮しなくてはならない。もちろん、フラッシュメモリの書換え時間によるシステム性能の低下を抑えることができる効果は大きい。

【0043】また、この時、キャッシュメモリ3が、図4に示すような4ウェイセットアソシエティブ方式を採用しているとすると、データが格納されるのは、3セット分だけである。残りの1セットは、CPUからの要求によって発生したミスヒットのライトアクセス後以外は、常に空き状態にしておくことになる。したがって、アクセス履歴情報の保持、更新は、3番目まで行えばよいことになる。

【0044】また、他に、フラッシュメモリにおいて一括消去の単位容量は、一回のアクセスデータと比べると、非常に大きいという問題がある。つまり、CPUの一回のアクセスデータは、数バイト程度であるのに対し、フラッシュメモリの消去の単位は、数百バイト以上である。したがって、リードアクセスのために、数バイトのデータをフラッシュメモリからキャッシュメモリに転送した後、そのデータの書換えが発生し、データをキャッシュメモリ上で書換えると、対応するフラッシュメモリ内の領域のデータを、いずれ書換える。このとき、フラッシュメモリ上では、数バイトだけの書換えはできないため、消去領域全てを別に設けた揮発性メモリに読み出し格納して、それに対して、該当個所の書換えを行い、再度、フラッシュメモリ内のデータの消去を行った領域にデータを書き込む、すなわち、フラッシュメモリにデータを書き戻すという処理が必要になる。

【0045】また、このとき、同一の消去領域内のデータがキャッシュメモリ内に別に存在するのであれば、同時に前書き戻しを行うのが効率的である。したがって、フラッシュメモリへのライトバック時には、処理が複雑になり、処理時間もかかってしまう。そこで、フラッシュメモリ内のデータを、キャッシュメモリに転送する際には、単位消去領域をまとめて転送することにし、この領域ごとにアクセス履歴を記録して、フラッシュメモリへのライトバックもこの領域ごとに行うことにすれば、フラッシュメモリ上における同一領域に存在する、キャッシュメモリ上のデータを、退避して書き込むという処理や、探すという処理は不必要となる。つまり、ライトアクセスが発生したら、そのデータを含む単位消去領域全てのデータをキャッシュメモリに展開するようにすればよい。

【0046】しかしながら、読み出し時には、数バイトのデータの読み出しのために、数百バイト以上を読み出さなければならないとすれば、非効率的である。そこで、リードアクセスのために、所望のデータがキャッシュメモリに転送されるときには、必要なバイト数だけキャッシュメモリに転送し、ライトアクセスのための転送時は、単位消去領域全てを転送することにすれば、リード動作もライト動作も効率的に行える。また、アドレスアレイに格納する情報として、キャッシュメモリ内のそのデータが書換えられたか否かを示すフラグを設定する構成にすることも、処理速度の向上等の大きな効果が期待できる。

【0047】次に、フラッシュメモリの応用例として、シリアルバッファ内蔵型の実施例について述べる。本発明において、フラッシュメモリの価格は、システムの価格を決定する重要な要素となる。不揮発性メモリで構成される主記憶は、大容量であることが要求されるため、主記憶を構成するフラッシュメモリは、安価である必要がある。メモリの価格を決定する大きな要因として、メモリ機能を実現する単位であるセルの大きさを表す「セル面積」がある。すなわち、ひとつの記憶セルが占める面積により、一定面積が有する記憶容量が定まるため、メモリチップとしての価格に大きく関係する。

【0048】フラッシュメモリにおいては、1セルの面積を可能な限り小さくするためには、ランダムアクセスを犠牲にする必要がある。つまり、ある容量を有している、チップ上の一ラインのデータを、一括してアクセスし、これをシーケンシャルにアクセス可能なシリアルバッファに転送し、当該シリアルバッファから、外部にシリアルアクセスして、データを出力する。したがって、ある一つのデータをアクセスするためには、同一ライン上のデータを一括してアクセスするためアクセス時間がかかり、結果的にランダムアクセス性能を落すことになる。しかし、この場合、連続したデータの要求に対しては一番目のアクセスは遅いものの、以降は高速にシリアルアクセスが可能である。通常、プログラム実行時は、アクセスの局所性から、高速なシリアルアクセスは非常に有効である。そこで、これを本発明に適用した実施例について説明する。

【0049】図6は、本実施例で用いるフラッシュメモリの構成例である。31は、フラッシュメモリチップ、32は、フラッシュメモリアレイ部、33は、シリアルバッファ部、34は、アクセスするアドレス、35は、シリアルクロック、36は、フラッシュメモリアレイ部32とシリアルバッファ33との間で転送されるラインデータ、37は、メモリチップから入出力されるデータである。ただし、ここでのアドレスとはラインアドレスであり、1ライン分全てのデータを転送するためのアドレスである。

【0050】次に、本実施例における動作を説明する。

まず、リードアクセスであれば、フラッシュメモリチップ31にアクセスするためラインアドレス34を入力すると、フラッシュメモリアレイ32上の該当するラインに存在するデータ36を、一括してシリアルバッファ33に転送する。転送されたデータは、その後シリアルクロック35を入力することによって、シリアルバッファ33から順番に取り出される。ライトアクセスの場合には、まず、シリアルクロック35の入力タイミングに合わせて、データを順にシリアルバッファ33に入力せしめ、データ入力が終了した後、フラッシュメモリアレイ32上の、ラインアドレス34で設定したラインに、データの転送を行う。以上が、本フラッシュメモリの動作の概要である。

【0051】本フラッシュメモリを使用した実施例を、図7、図8および図9を参照して説明する。図7は、ハードウェア構成を示し、図1とほぼ同様の構成である。図1と同一の符号が付加されている構成要素は、同一の機能を有する手段である。ただし、コントローラ6の構成は、後述するが、若干異なっている。図1のフラッシュメモリ2の部分は、図6にて説明したフラッシュメモリチップ31を使用している。図8は、本実施例におけるコントローラ6の構成例を示したものであり、38は、シリアルバッファ33用クロック発生回路で、他の構成要素は図2にて説明した構成要素と同一である。図9は、コントローラ6の動作を示すフローチャートであり、CPU1のアドレスと、フラッシュメモリ31のアドレスの扱いの違いが問題となる。

【0052】さて、図6、図7、図9を参照して動作を説明する。まず、CPU1からのアクセス要求があり（ステップa）、アドレスヒットであればキャッシュメモリ3をアクセスし（ステップb）、アドレスミスヒットであれば、キャッシュメモリ3内に新たにデータを格納するための新規格納領域を作り出す（ステップc）。そして、フラッシュメモリ31をアクセスするためのアドレスを含むラインのアドレス34を与えて、シリアルバッファ33に、そのラインデータを転送して、シリアルクロック35を入力して、データを引き出す（ステップd）。この引き出されたデータを、キャッシュメモリ3の新規格納領域に格納して（ステップe）、さらに、リードアクセスの場合には、この新規格納領域の中からCPU1のメモリバス7上に、所望のデータを出力する（ステップf）。一方、ライトアクセスの場合には、新規格納領域内のアクセスするアドレスに該当する個所にデータの書き込みを行なう（ステップg）。

【0053】最後に、アドレスアレイの内容を更新して処理を終了する（ステップh）。ここで、ミスヒット時の処理性能向上のためには、先の実施例である図5の動作と同様、新規格納領域を作る処理をCPU1のアクセス終了後に行い、新規格納領域を、実際に書き込み要求がある以前に作成しておくといふ。なお、本実施例にお

いては、アドレスアレイやアクセス履歴の記録は、全てライン単位に設定するものとする。また、このラインアクセスのフラッシュメモリに対して、セットアソシエティブ方式を適用すると、効率的にハード構成の簡略化が図れる。次の実施例は、シリアルバッファを内蔵した構造であるとともに、シリアルアクセス開始アドレスが設定可能なフラッシュメモリチップを使用したものである。詳しくは、まず、所望のラインをアクセスするために、ラインのアドレスを入力する。それとともに、そのラインデータがシリアルバッファに転送された後、シリアルクロック入力により出力される最初のデータを、ライン内の何個目のデータとするかを、アドレス入力により設定可能とするものである。

【0054】例えば、1データの長さが1バイト、1ラインが512バイトとして、512個のデータのうちの128番目のデータが必要であれば、ライン自体のアドレスおよびライン内の128番目というアドレスを指定する。これにより、シリアルクロック入力後の最初の出力データは、128番目のデータとなり、所望のデータが瞬時にアクセス可能となる。かかるアクセスが可能であると、ラインの途中で存在するデータを取り出すとき、ライン上の何番目に存在するかに相当する個数のシリアルクロックの入力が不要となり、ラインアクセス構造でありながら、ランダムアクセスを比較的高速に行うことが可能となる。このようなメモリを使用したときの実施例を、図10、図11を参照して説明する。

【0055】図10は、リードアクセス時のデータの扱いを、図7、図8における実施例と対比して示したもので、図中、3は、キャッシュメモリ、33は、図6の構成のフラッシュメモリ31のシリアルバッファ33の部分であり、シリアルクロック35の入力によってデータの入出力が可能である。そして、特に、図10（2）に示すシリアルバッファ33については、シリアルバッファ33内の先頭アドレスの設定が可能なものとする。41は、キャッシュメモリ内部の転送先領域であり、シリアルバッファ33と等しい容量を有している。42は、シリアルバッファ33の中で先頭に指定されたアドレスから1ないし数バイト離れた領域を示しており、その容量は、キャッシュメモリ3のデータ格納単位と等しいものとする。この容量は、CPU性能、キャッシュメモリ容量などにもとづいて決定される。43は、シリアルバッファ33上の転送データ42の転送先であり、キャッシュメモリ3の領域を、格納単位で区切って作成した複数の領域のうちの一つである。

【0056】図10（1）においては、フラッシュメモリアレイからシリアルバッファ33に転送されたデータを、1ライン分そのままキャッシュメモリ3に転送して格納する。つまり、図9で示したフローチャートに従ってデータを扱っている。このデータ量は、フラッシュメモリの構成、具体的には、シリアルバッファの容量に依

存する。フラッシュメモリは、格納されているデータを、ある容量単位で一括消去する構造にしたために、メモリ構造の高集積化が可能となっているが、シリアルバッファの大きさは、一括消去の単位容量に依存しているため、ある程度の大きさになることは、フラッシュメモリの特性のため避けることはできない。

【0057】現在、この一括消去の容量を、HDD (Hard Disk Drive) におけるセクタの容量にあわせ、512バイトとするものが開発されている。このようなメモリを採用したとすれば、図10の(1)では、データ転送単位が512バイトとなる。一方、図10(2)では、この容量をさらに細かく区分してデータの転送を行う。したがって、例えば数バイトの小さい単位でのデータの転送も可能であるため、転送時間を向上できる。また、キャッシュメモリ3には、小さい単位で多くの個所に分割して格納個所を構成しているため、長いシーケンシャルアクセスがあまり発生しないプログラムの実行時には、ヒット率を高くすることが期待できる。もちろん、シーケンシャルなアクセスに対しては、単に、連続アドレスで格納すればよいので、どちらでも対処可能である。

【0058】この図10(2)に示す動作を説明するためのフローチャートを、図11に示す。まずCPU1からアクセス要求が出される(ステップa)。次に、アドレス値をアドレスレイ4で比較して、アドレスヒット、ミスヒットを判断する(ステップb)。アドレスヒットの場合には、キャッシュメモリ3をそのままアクセスする。ミスヒットの場合には、アクセスするアドレスを含む1ラインを、フラッシュメモリアレイ32からシリアルバッファ33に転送する(ステップc)。そして、リードアクセスの場合には、アクセスデータのライン内における先頭アドレスを設定する(ステップd)。そして、シリアルバッファ33にシリアルクロック35を入力して所望のデータを取り出す(ステップe)。次に、このデータをキャッシュメモリ3に用意されている新規格納領域に格納する(ステップf)。CPU1は、キャッシュメモリ3をアクセスし、アクセスされたデータは、メモリバス7上に出力される(ステップg)。

【0059】ライトアクセスの場合には、ステップcでシリアルバッファ33に転送したデータを、全てキャッシュメモリ3に転送し(ステップh)、キャッシュメモリ3上に転送したデータに対して、アクセス個所をCPU1からのライトデータに書換える(ステップi)。そして、次のミスヒットアクセスに備え、キャッシュメモリ3内に新規格納領域を作り出す(ステップj)。最後に、アドレスレイ4に格納されている情報を、その時の状態に合致するよう更新する(ステップk)。ステップjにおける、新規格納領域を作る処理は、キャッシュメモリ3の単位格納領域がリードアクセスされた場合には、データをライトバックする必要はないため、単にデ

ータを前記単位格納領域に書きすればよい。ライトアクセスの場合には、ライン単位で処理を行うため、候補のラインを決定し、そのラインに存在するデータをシリアルバッファ33に格納して、フラッシュメモリアレイ32に転送するという処理が必要である。

【0060】なお、DRAMとSRAMの組合せによる、情報処理装置で搭載されているキャッシュシステムにおいては、数バイトから数10バイトを1ブロック容量として、キャッシュ効率を上げている。同様に、本実施例においてもCPUのワード単位ではなく、16バイトや64バイト等の単位で、キャッシュメモリとフラッシュメモリとのデータ転送を行うことにより、制御の簡単化、ヒット率の向上に効果的となる場合がある。大容量不揮発性の主記憶を用いたシステムで、CPUが何らかの原因で暴走した場合の対処について説明する。かかる事態に対処するために、例えば、フラッシュメモリの消去単位ごとに、書き込み禁止フラグを格納する格納レジスタを設け、通常書き変わらないデータに対しては、書き込み禁止フラグを立てる。書き込み禁止領域の書換えにおいては、先述の書き込み禁止フラグを下げ、CPUに割込みが発生しないようにしてから、データの書換えを行うのが正規の方法とする。フラグが立ったままでこの領域への書き込みが発生した場合には、CPUに対する割込みを発生し、ユーザに対し確認を得るようにする。確認方法として、例えば出力手段61のCRT等のディスプレイに、CPUがメッセージを表示する処理を行うようにしておけばよい。

【0061】つまり、ユーザはこのときに、例えば、表示メッセージによって、暴走状態にあることを認識でき、キャッシュメモリ内のデータのうち、退避すべきものを選択して、退避することができ、また、CPUが備えるプログラムカウンタのリセットを実行できる。また、キャッシュメモリにおいても、アドレスレイ内に同様のフラグレジスタを設けておけば、キャッシュメモリ内のデータに対する安全性が向上する。以上の処理を実現する装置の構成図を図12に示した。図中、1は、CPU、3は、アドレスレイ、6は、コントローラ、60は、入力手段、61は、出力手段、51は、フラッシュメモリの消去単位ごとに設けてある冗長領域、52は、冗長領域51に設定した書き込み禁止フラグ、53は、アドレスレイ内に設けた書き込み禁止フラグ、54は、CPUへの割込み要求信号であり、マスク不可能な割込み信号が望ましい。

【0062】入力手段60は、少なくともデータをアクセスするためのコマンド等の各種コマンド、後述するメッセージに対処するための指示等を入力する機能を有する入力手段である。例えば、マウス、キーボード等によって実現される。出力手段61は、後述するメッセージや、その他の必要な情報を出力する機能を有する手段であり、例えば、CRT、ELディスプレイ、液晶ディス

プレイ等の表示手段により実現できる。

【0063】コントローラ6は、フラッシュメモリ31やキャッシュメモリ3への書き込み要求が発生すると、その個所の書き込み禁止フラグ52、53を参照し、このフラグが立っている場合には、割込み要求信号54を発生する。CPU1は、これを受けると無条件に、ユーザにこの状況に対する処理を求めるシーケンスを実行する。ユーザに求める処理の内容としては、以下の内容が考えられる。(1)書き込みを行うか、(2)暴走状態と判断するか、(3)キャッシュメモリから主記憶へ退避するデータの指定、(4)プログラムカウンタをクリアするか、(5)キャッシュメモリ内をクリアするか、などである。ユーザに対する問い合わせは、例えば、CRT等の表示装置に、その旨をメッセージ表示する構成にすれば良い。また、ユーザは、問い合わせに対応するために、マウス、キーボード等の入力装置を介して、必要な指示を与えるような構成にすればよい。また、CPU1は、予め格納されたプログラムに従って、指示に対応した処理を行うように構成しておけばよい。以上の処理により、暴走による重要データの喪失を、ある程度回避できるようにする。

【0064】以上の全ての実施例においては、キャッシュメモリ3は、バスを介してCPU1に直接接続されている。しかし、情報機器によっては、CPU1とキャッシュメモリ3の間に、メモリバスコントローラや、SRAMによるキャッシュシステム等の装置が挿入されるものもある。ここでは、CPU1がそれらを含めた構成のものであるとする。キャッシュメモリ3が、情報処理システムに備わるメモリバスに、直接接続されることが重要な点である。また、作業中断による電源遮断の際には、キャッシュメモリからフラッシュメモリにデータを退避する必要がある。キャッシュメモリ内のデータのうち、リードアクセスだけであったために書き変わっていないデータに関しては、データ退避の必要はない。しかし、ライトアクセスにより書き換えられたデータは、キャッシュメモリ上には存在するが、フラッシュメモリ上には存在しないため、退避を行わないと、最新のデータが失われることになる。

【0065】また、いわゆるリジューム機能を実現するためには、退避の際に、CPUが備える各レジスタが格納する情報や、表示装置、入出力装置等が備えるデータ格納部内に存在する内部データも記憶しておく必要がある。すなわち、リスタートに必要な、データを全て退避しておかなければ、リジューム機能を実現できない。このため、電源遮断を機械的なスイッチにより行うのではなく、電源遮断情報を論理的に処理するシーケンスを実行する手段が必要である。このような手段は、CPU、ROM、RAM、各種CMOS等を有して実現可能である。この場合、必要な処理を、CPUが行うためのプログラムをROMに内蔵しておけばよい。また、キャッシ

ュメモリ内に格納されたデータが喪失されないようにするために、定期的に、キャッシュメモリのデータをフラッシュメモリにライトバックする処理を行う構成にしておくことが効果的である。また、キャッシュメモリとして、バックアップ電源付きのSRAMや強誘電体メモリ(FRAM: Ferroelectric RAM)を使用すると、高速ランダムアクセス可能でありながら、データを喪失することがなくなる。

【0066】以上のように、本発明によれば、データの書換え速度が遅いフラッシュメモリを主記憶としたシステムを構築でき、バッテリーによるバックアップがなくとも、リジューム機能やクイックスタート機能が容易に実現できる。また、DRAMより安価になり得るフラッシュメモリを主記憶とするため、大容量の主記憶を安価に提供できる。さらに、不揮発性の記憶手段によって主記憶を構成したことにより、磁気ディスク装置や、その他の外部記憶装置が不要となり、システムの簡略化が図れる。さらにまた、処理途中でのファイルアクセスが高速に行え、システムの性能が向上する。加えて、本発明においては、フラッシュメモリには、ランダムアクセス性能を主眼にしたNOR型だけではなく、ラインアクセス方式のNAND型、AND型等、各種のフラッシュメモリを採用することができることが大きな特徴である。

【0067】図13に次の実施例の概略図を示す。図13において、101はCPU、102はバス、300はコピーバック方式のキャッシュメモリシステム、104はmバイト(mは整数)を1ブロックとするフラッシュメモリで構成される記憶装置(主記憶)、105は制御回路である。キャッシュメモリシステム300内には、アドレス情報を保持するアドレスレイ310、データを保持するキャッシュメモリ320、およびアドレスレイ310のアドレスとCPU101からのアドレスを比較するアドレス比較器330で構成される。データを保持するキャッシュメモリ320はn本(nは整数)のmバイト単位のレジスタ321で構成される。アドレスを保持するアドレスレイ310は、各々、アドレス情報を保持する部分(アドレス部)aと、保持されているアドレス情報に該当する記憶装置104内のブロックが消去済みが否かを示す情報を保持する部分(消去情報部)bと、アドレスレイ310内の該当するレジスタ321が更新されたか否かの情報を保持する部分(更新情報部)cとからなるn本のレジスタ311で構成される。更新情報部cは既存のフィールドであるが、消去情報部bは、本発明で新たに設けたフィールドである。312、322はアドレスレイ310、キャッシュメモリ320の制御信号である。

【0068】図14は、図13の制御回路105の処理フローを示す図である。以下、図14に示す処理フローについて説明する。CPU101から記憶装置104へリードアクセスが発生した場合(STEP1)、比較器

330からのヒット判定信号331でキャッシュヒットを判定すると(STEP2)、キャッシュメモリ320内のレジスタ321の該当箇所からバス102へデータを出し(STEP21)、CPU101はそのデータを読み込む。

【0069】CPU101からのリードアクセスでミスヒットを判定した場合で、更に、置き換えアルゴリズムによって選択されたレジスタ311内の更新情報部cの情報から、キャッシュメモリ320内のレジスタ321の該当箇所が更新されていると判明した場合(STEP22)は、STEP226へ進む。ここでレジスタ321から記憶装置104内の該当するブロックへデータを書き込む。すなわちキャッシュメモリにおいて更新されたデータを記憶装置104の該当ブロックへ書き戻す処理を行う。続いて、CPU101からアクセスされたデータを含む記憶装置104内のブロックデータを、レジスタ321内の置き換え対象になった個所に書き込む。そして、当該CPUアドレスをレジスタ311へ書き込むとともに、レジスタ311の更新情報部cに未更新を示す情報を書き込む(STEP227)。そしてCPU101からアクセスされたデータをレジスタ321からバス102に出力する(STEP228)。STEP228の処理が完了するとSTEP1に戻り、CPU101からの次のアクセスを待つ。尚、ここで言うレジスタ321が更新されている状態とは、記憶装置104内の該当するブロックのデータとレジスタ321内に格納されているデータが異なる、つまりレジスタ321だけが書換えられている状態を示す。本明細書では以下同様である。

【0070】リードアクセスでミスヒットを判定し、更に新たに選択されたレジスタ311内の更新情報cの情報から、キャッシュメモリの置き換えアルゴリズムに従って選んだ、キャッシュメモリ320内の書き戻すべきレジスタ321が更新されていないことを認識すると(STEP22)、レジスタ321のデータを記憶装置104へ書き戻す処理(STEP226)を省略する。そして記憶装置104内の該当するブロックの全データをレジスタ321の選択された個所に書き込みとともに、レジスタ311にアドレスを書き込む(STEP227)。また、STEP227でレジスタ311の更新情報部cに未更新を示す情報を書き込み。そしてCPU101からアクセスされたデータをレジスタ321からバス102に出力する(STEP228)。STEP228の処理が完了するとSTEP1に戻り、CPU101からの次のアクセスを待つ。CPU101から記憶装置104へライトアクセスが発生した場合(STEP1)、キャッシュメモリがヒットすると(STEP3)、レジスタ321の該当個所にCPU101からの書き込みデータを書き込むとともに、レジスタ311を更新する(STEP324)。そして、STEP324

の処理が完了すると、書き込み対象ブロックが消去済みかどうかを判定して、消去済みでなければ該当ブロックの事前消去処理を起動する(STEP33)。その後、STEP1に戻りCPU101からの次のアクセスを待つ。

【0071】STEP3でキャッシュミスヒットを判定し、置き換えアルゴリズムで選択されたレジスタ311の更新情報部cの情報から、レジスタ321に保持されているデータが更新済みであることを認識すると(STEP31)、レジスタ321内から選択した置き換えデータを記憶装置104内の該当ブロックへ書き戻す(STEP322)。このSTEP322では、レジスタ311内のブロック消去情報部bに未消去を示す情報を書き込む。STEP322の処理が完了すると、CPU101からのアクセス個所を含む記憶装置104内のブロックデータを、置き換え対象になった個所に書き込み、さらにレジスタ311に当該アドレスを書き込む(STEP323)。そしてレジスタ321の該当する個所に、CPU101からの書き込みデータを書き込む(STEP324)。またSTEP324において、レジスタ311内のレジスタ更新情報部cに更新済みを示す情報を書き込む。一方、STEP31でレジスタ311の更新情報部cの情報からレジスタ321に保持されているデータが更新されていないことを認識すると、データの書き戻し(STEP322)は不要なので、これを省略しSTEP324に進む。以下、STEP324の処理が完了すると、書き込み対象ブロックが消去済みかどうかを判定して、消去済みでなければ該当ブロックの事前消去処理を起動する(STEP33)。その後、STEP1に戻りCPU101からの次のアクセスを待つ。

【0072】図14のフローにおいて、特徴的な点は、STEP323、324でレジスタ311、321を更新した後に、記憶装置104の対応するブロックを前もって消去しておく(STEP33)にある。この場合、本来ならば、コピーバック方式のキャッシュメモリにおいては、該当レジスタ311、321を更新すれば足りる。しかし、この更新に伴い、更新情報部cが更新済みとされるため、このレジスタ321の内容はいずれ記憶装置104の対応ブロックに書き戻される。その際、書き戻しに先立って、そのブロックの消去が必要となる。本実施例によれば、その、いずれは消去される対応ブロックを予め消去しておくので、次回以降のライトアクセス時のキャッシュミスヒットの際の、記憶装置104の新たなブロックへの書き戻し(STEP322)において、本来ならこれに先立つべき消去処理が省略される分高速になる。

【0073】なお、本実施例は、CPUのアクセスサイズと、フラッシュメモリの消去ブロック単位容量が異なる場合を示している。たとえば、CPUは主記憶に対し、16バイトのバースト転送によるデータのやり取り

を行なっているのに対し、フラッシュメモリの消去ブロック単位容量は512バイトであった場合などを想定している。フラッシュメモリの消去ブロック単位容量は、メモリの高集積化を図るため、CPUのアクセス単位より非常に大きいのが一般的である。ただし、もしこれらのサイズをあわせたシステムの場合であれば、ライトミスヒット時の置き換え処理の際、図14におけるSTEP323の処理において、CPU101のアクセス箇所を含む記憶装置104内のブロックデータを、置き換え対象になった個所に書き込む処理が不要になる。つまり、CPU101のライトデータを、書き戻しを行なった個所にそのまま書き込むだけでよい。

【0074】図19により、本実施例による効果を説明する。ここでは、ブロック消去時間を10ms、ブロックへの書き込み時間を4ms、キャッシュメモリ320の更新時間を100nsと想定した場合の従来の方式と本実施例の方式におけるアクセス時間を比較する。今、CPU101から記憶装置104へのライトアクセスが発生し、アドレスがミスヒットした場合を想定する。従来であれば、この時点から、選択されたデータレジスタのデータ書き戻し先のブロック消去(10ms)を開始し、その後、キャッシュメモリ300の該当レジスタ321のデータを記憶装置104の対応ブロックへ書き戻す処理(4ms)を行い、さらに、キャッシュメモリ300を更新する(100ns)。これでライトアクセスが完了し、CPU101は次の処理を開始する。従って、ライトアクセスにトータル約14msを要する。

【0075】これに対し、本実施例では、アドレスがミスヒットした時点で、選択されたデータレジスタが更新済み(書き戻し要)であっても、書き戻し先の対応ブロックの消去が既に終了している。したがって、直ちに該当ブロックへのデータ書き戻し処理(4ms)を開始することができる。したがって、本実施例の場合のライトアクセスはトータル約4msであり、従来の方式に比べて70%程度速く、CPU101が次の処理に移行できることがわかる。また、次の処理として、上述のキャッシュメモリ300上で更新されたデータを書き戻す際の、対応ブロックの消去処理を行なっておく。図15に、CPU101から記憶装置104へのアクセスデータのサイズが記憶装置104のブロックサイズより小さい場合の実施例の概略図を示す。図15において101、102、300、310、312、320、321、322、330、331、104、401、105は図13の実施例と同様である。

【0076】313は、アドレス情報を保持するアドレス情報部aと、キャッシュメモリ320内の該当するレジスタ321が更新されたか否かの情報を保持する更新情報部cと、レジスタ321内の更新された領域を示す情報を保持する更新領域情報部dからなるレジスタである。350は、記憶装置104への書き込みの際に用い

るアドレス情報用レジスタ、340は、記憶装置104への書き込みの際に用いるmバイト(記憶装置104の1ブロック分)のデータ用レジスタである。アドレス情報用レジスタ350は、アドレスアレイ310の各レジスタ313に対応してn本設けられる。同様に、データ用レジスタ340は、キャッシュメモリ320の各レジスタ321に対応してn本設けられる。図16に、図15の制御回路105の処理フローを示す。以下、この処理フローについて説明する。尚、この処理では、図13の実施例のブロック消去の動作については、アクセスデータサイズとブロックサイズの違いに対する対処についてのみ説明する。ブロック消去動作を考慮した処理フローについては後述する。

【0077】CPU101から記憶装置104へリードアクセスが発生した場合(STEP1)、比較器330からのヒット判定信号331でキャッシュヒットを判定すると(STEP2)、アクセスされたレジスタ321内の該当する部分のデータをバス102へ出力し(STEP21)、CPU101はそのデータを読み込む。同様にリードアクセスでミスヒットを判定した場合(STEP2)は、キャッシュメモリの置き換えアルゴリズムで選択されたレジスタ313内の更新情報部cの情報から、キャッシュメモリ320内の書き戻すレジスタ321が更新されていない場合は(STEP22)、CPU101からの読みだしアドレスに該当する記憶装置104内の対応するブロックの全データをレジスタ321に書き込む処理を行なう(STEP224)。尚、ここで言うレジスタ321が更新されていない状態とは図13の実施例同様、レジスタ321内に格納されているデータと記憶装置104内の対応するデータとが同じである状態を示す。また、STEP224でレジスタ311内の更新情報部cに未更新を示す情報を書き込む。STEP224の処理が完了すると、CPU101が要求するレジスタ321内のアクセスされたデータをバス102へ出力する(STEP225)。

【0078】STEP22で書き戻すべきレジスタ321が更新されていることを認識すると、レジスタ313のアドレス情報に対応する記憶装置104内のブロックの全データを読み出してレジスタ340に書き込む(STEP221)。STEP221が完了すると、レジスタ313内のdの情報をもとにレジスタ321内の更新されたデータのみをレジスタ340の該当部分に書き込む処理と、レジスタ313内のアドレス情報をレジスタ350に書き込む処理を行なう(STEP222)。STEP222が完了すると、該当レジスタ350のアドレス情報に対応する記憶装置104内ブロックを消去し、該当レジスタ340内の全データを、消去したブロックに書き込む(STEP223)。STEP223が完了するとSTEP224、STEP225へと進む。STEP225の処理が完了すると、STEP1に戻り

31

CPU101からの次のアクセスを待つ。

【0079】CPU101から記憶装置104へライトアクセスが発生した場合（STEP1）、キャッシュメモリがヒットすると（STEP3）、STEP314へ進みCPU101からの書き込みデータをレジスタ321内の該当する部分へ書き込む。また、レジスタ313内の更新情報部cに更新済みを示す情報を、更新領域情報部dにレジスタ321の更新された領域を示す情報を書き込む。STEP3でキャッシュミスヒットを判定し、レジスタ313の更新情報部cから、書き戻すべきレジスタ321が更新されていないことを認識すると（STEP31）、STEP314へ進み前述した処理を行なう。尚、ここで言うレジスタ321が更新されていない状態とは図13の実施例同様記憶装置104内の該当するデータとレジスタ321内に格納されているデータが同じである状態を示す。STEP31で記憶装置104内の書き戻すべきレジスタが更新済みであることを認識すると、レジスタ313のアドレス情報に対応する記憶装置104内のブロックの全データを読み出してレジスタ340に書き込む（STEP311）。STEP311が完了すると、レジスタ313内の更新領域情報部dの情報をもとにレジスタ321内の更新されたデータのみを該当レジスタ340の該当部分に書き込む処理と、レジスタ313内のアドレス情報を該当レジスタ350に書き込む処理を行なう（STEP312）。STEP312が完了すると、該当レジスタ350のアドレス情報に対応する記憶装置104内ブロックを消去し、該当レジスタ340内の全データを消去したブロックに書き込む（STEP313）。STEP313の処理が完了すると、STEP314に進み、前述した処理を行なう。STEP314が完了すると、STEP1に戻りCPU101からの次のアクセスを待つ。

【0080】図15の実施例によれば、CPU101からのライトアクセスが記憶装置104のブロックサイズより小さい場合でも、書き戻すブロックのデータを一旦該当レジスタ340に取り込み、ブロック内の新たに書き込まれる部分のみを該当レジスタ340上で更新し、該当レジスタ340のデータ全てを一括して記憶装置104内の対応ブロックに書き込むことにより、ブロック内の部分書き込みが実現できる。また、CPU101からのリードアクセスが記憶装置104のブロックサイズより小さい場合でも、ブロック内の全データを一旦レジスタ321に取り込み、レジスタ321内の該当するデータのみをバス102へ出力することにより、ブロック内の一部のデータを読み出すことができる。図20は、図16の処理フローにおいて図14の事前のブロック消去を考慮し、両者のステップを組み合わせたものである。従って、図15におけるレジスタ313には、図13で示した消去情報部bが追加される。

【0081】図20中、図14および図16と同様のス

32

テップには同じ参照符号を付してある。図16のSTEP223がSTEP801に置き換えられ、図14のSTEP322および図16のSTEP313がSTEP802に置き換えられている。また、図14のSTEP33の前にSTEP803が新たに追加されている。これにより、CPU101からのライトアクセスが記憶装置104のブロックサイズより小さい場合に、事前のブロック消去を行うことができる。STEP3においてキャッシュメモリがヒットし、該当するレジスタ313内の消去情報部bの情報が、ヒットしたメモリライトアクセスに対応する記憶装置4内の対応するブロックが未消去であることを示すため、STEP803へ進む、というフローは、次の2通りの状況において考えられる。

【0082】1つは、ヒットしたメモリライトアクセスにより更新される前に、ミスヒットリードアクセスによりSTEP224、STEP225という処理フローを行うことによって、データが記憶装置104内のブロックから読みだされ、レジスタ321内に格納された場合に起こる。この場合はリードされた記憶装置104内の該当ブロックは消去されていない。そのため、この状態で全く同じアドレスに対しライトアクセスが発生するとキャッシュヒットメモリライトアクセスとなりSTEP3からSTEP803へ進む。もう1つは、電源OFFから電源ONに変化した後などにより、記憶装置104内のデータのみ有効で、キャッシュメモリ内のレジスタ321、313、340、350内のデータが無効な状態、すなわち空きの状態の時に、記憶装置104へのメモリライトアクセスが発生した場合である。このとき、レジスタのリプレースは起きないため、キャッシュヒットとして扱える。そして、対応する記憶装置104内の該当ブロックを消去するだけで更新は省略し、該当するレジスタ321のみ更新するので、STEP3からSTEP803へ進む。

【0083】次に、電源を再供給後に、しばらくミスヒットが続く著しい性能劣化を起こす、という問題解決の実施例を、図21、22、23を用いて説明する。図21は、フラッシュメモリ2内に、アドレスアレイデータを退避するための領域を確保したことを示す図である。図中、既出の番号は、以前に説明したものと同一のものである。新しい構成要素を説明する。201は本情報処理装置の電源スイッチであり、機械的に電源を遮断するものではなく、本情報処理装置の電源供給装置204を通して、CPU1に電源遮断要求信号202を出力する働きを持つものである。202はCPU1に働きかける電源遮断要求信号であり、CPU1はこれを受けると、電源遮断のルーチンを実行することになっているものとする。203はフラッシュメモリ2内のある領域に設けられた、アドレスアレイのデータの退避領域である。205はCPU1から電源供給装置204に出力される電源遮断許可信号である。電源遮断許可信号205は、

情報処理装置の動作中は、常にネゲートされており、電源スイッチ201が「切」になり、CPU1が電源遮断ルーチンを終了したらアサートされ、このとき電源供給装置204は、電源供給を遮断するように動作するものとする。図22は、CPU1が電源遮断要求信号202を受けた後に実行される、電源遮断ルーチンである。

【0084】図21を参照しながら、図22のフローを追って、本実施例の動作を説明する。CPU1は何らかの処理を行っている最中に、電源遮断要求信号202を受け取ったものとする。まず、現在行っている処理が、外部デバイスと関連した処理であること等により、処理の中断ができない、という場合を除き、処理を中断する(ステップa)。処理の中断ができない処理の途中であった場合は、中断しても構わない状態まで処理を続ける。基本的にはあらゆる状態においても処理を中断すべきとした方が好ましい。次に、キャッシュメモリ3に格納されているもののうち、CPU1からのライトデータについて、アドレスアレイ4を参照して、フラッシュメモリ2の該当する領域に書き戻す(ステップb)。もしライトデータとリードデータを区別していない、あるいは処理の簡略化を図るのであれば、キャッシュメモリ3内の全てのデータを書き戻してもよいが、時間がかかってしまうため、ライトデータのみの書き戻しの方が好ましい。次に、アドレスアレイ4に格納されている、キャッシュメモリ3とフラッシュメモリ2のデータの対比を示すアドレスデータを、フラッシュメモリ2の領域203に退避する(ステップc)。以上の処理が終了したら、CPU1は電源供給装置204に電源遮断許可信号205を出力する(ステップd)。これを受けて電源供給装置204は、本情報処理装置への電源供給を停止する(ステップe)。

【0085】一方、図23は電源供給開始時のフローである。同様に図21を参照しながら説明すると、まず、電源スイッチが「入」にされると、無条件に電源供給装置204は、本情報処理装置に電源を供給するよう動作する(ステップf)。もちろん電源供給装置が電源供給可能な状態である場合に限る。次にCPU1は情報処理装置に必要な初期処理を実行する(ステップg)。例えば各デバイスの動作チェック、内部レジスタへの設定などである。次に、フラッシュメモリ2のアドレスアレイデータ退避領域203をアドレスアレイ4に転送する(ステップh)。次にアドレスアレイ4を参照して、電源遮断直前にキャッシュメモリ3に格納されていたデータを、フラッシュメモリ2内の該当する領域から引出し、キャッシュメモリ3の該当する領域に格納する(ステップi)。以上の動作によりキャッシュメモリは電源遮断前の状態に簡単に復帰することができ、以降のCPU1からのアクセスに対し、高いヒット率でのアクセスを提供することができる。なお、以上の実施例では、ア

ドレスアレイ4を揮発性のメモリとした場合のものであるが、アドレスアレイ4を不揮発性メモリとすれば、アドレスアレイ4に格納されているデータの退避処理は必要なくなり、電源再立ち上げ後の処理においても、アドレスアレイ4の格納データの復帰処理は必要なくなる。この場合、アドレスアレイ4は、ランダムアクセス(リードライトとも)可能であることが好ましいため、バックアップ電源付きのSRAMや、強誘電体メモリ(FRAM)を用いるべきである。

【0086】次は、電源の突発的な遮断が起きたときに被害を小さくする方式の実施例について説明する。キャッシュメモリに格納されたデータのうち、ライトアクセスにより書き込まれたデータは、フラッシュメモリにライトバックされていないうちは、電源遮断により失われては困るデータである。そのため、電源遮断時には、このデータをフラッシュメモリに格納する必要がある。そして、電源遮断時にデータ退避を行うためには、電源遮断要求を、実際の電源遮断実行前に認識し、フラッシュメモリへの格納を行った後に電源遮断を行うことになる。しかし、電源が突発的に遮断された場合には、このフラッシュメモリへの格納動作が不可能であるため、フラッシュメモリにライトバックされていないキャッシュメモリ上のデータは、失われてしまうことになる。先の実施例で行ったアドレスアレイのデータ退避についても同様のことがいえる。そこで、少しでもこのような事故を防ぐために、定期的にキャッシュメモリのデータをフラッシュメモリにライトバックする処理を行う。

【0087】これを実現するための構成を図24に示した。図中、211は定期的な時間を図るためのタイマ、212は一定時間ごとにCPU1に入力される、タイマ割込み信号である。なお、そのほかの既出の番号は、先述の説明の通りである。211のタイマは、一定時間ごとにタイマ割込み信号212を発生し、CPU1に入力する。これを受けたCPU1は、キャッシュメモリ3において、フラッシュメモリ2にライトバックしていないライトデータを、フラッシュメモリ2の該当する領域に書き込む。

【0088】タイマ211が計測する一定時間についてであるが、この時間が短いと、フラッシュメモリ2の書換えが何度も発生し、フラッシュメモリ2を劣化することになる。逆に長いと、突発的な電源の遮断が起きたときに、データを失う確率も、失うデータ容量も、大きくなることになる。したがってフラッシュメモリを破壊しない程度でできる限り短い時間間隔にすべきである。この時間間隔は、フラッシュメモリの書換え可能回数と、この情報処理装置の寿命により決定するのが好ましい。その算出式を(式1)に示した。

【0089】

(情報処理装置の使用時間)

(時間間隔) \geq

(式1)

(フラッシュメモリの書換え可能回数)

例えば、一日平均8時間使用して、情報処理装置の目標寿命を5年とし、フラッシュメモリの書換え可能回数を100万回とすれば、約53秒ごとに退避することができる。また、このとき同時にアドレスアレイの格納データも退避すれば、アドレスアレイを揮発性メモリとした場合でも、突然の電源遮断でデータを失われる確率が小さくなり、再立ち上げ直後のヒット率の低下を回避することができるようになる。なお、本実施例を実施していても、電源遮断時に最終的なデータを退避すべきであることは当然である。

【0090】

【発明の効果】情報処理装置の主記憶としてフラッシュメモリを採用することにより、バックアップ電源なしでリジュームやクイックスタートが可能で、かつ、安価な大容量主記憶を持った情報処理装置が構成できる。

【0091】また、電源遮断時の退避処理を行い、プログラムが暴走したときに備えたライトプロテクト手段を持つことにより、データの安全性を増すことができる。

【0092】また、本発明では、コピーバックキャッシュメモリの内容更新の際に、いずれは消去される記憶装置ブロックを事前に消去しておく。そのため、ブロックへの書き戻しが発生した場合の消去処理が不要である。直ちに書き込み処理を開始することができる。従ってフラッシュメモリを用いた記憶装置へのアクセス時間を短縮できる。

【0093】また、キャッシュメモリのデータを、一旦電源を遮断し、再開するときに復帰できる手段を提供するので、再開後のヒット率を向上させることができる。

【0094】また、キャッシュメモリのデータあるいはアドレスアレイのデータを、定期的にフラッシュメモリに書き込むことにより、突然、電源供給が停止しても、失うデータをなくすか、あるいは少なくすることができるので、被害を最小限に止めることができる。

【図面の簡単な説明】

【図1】フラッシュメモリを主記憶とした情報処理装置の一実施例の概略の構成図である。

【図2】フラッシュメモリを主記憶としたシステムのコントローラの構成図である。

【図3】フラッシュメモリを主記憶としたシステムのコントローラの動作フローチャートである。

【図4】セットアソシエティブ方式のキャッシュメモリ周辺の構成図である。

【図5】ミスヒット時に高速化を図るためのコントローラの動作を説明するフローチャートである。

【図6】ラインアクセス方式のフラッシュメモリの動作説明図である。

【図7】本発明にかかるフラッシュメモリを使用した主

記憶システムの構成図である。

【図8】本発明にかかるフラッシュメモリを使用した主記憶システムのコントローラの構成図である。

【図9】コントローラの動作を説明するフローチャートである。

【図10】ラインアクセスのフラッシュメモリのシリアルバッファとキャッシュメモリ間のデータ転送の説明図である。

【図11】シリアルバッファ内のアドレス設定可能なフラッシュメモリを使用した場合のコントローラの動作を説明するフローチャートである。

【図12】書き込み禁止手段の実現手段の構成例の説明図である。

【図13】本発明における書き戻し処理の高速化を図る一実施例を示す概略図である。

【図14】図13の制御回路105の処理フローを示す図である。

【図15】本発明における書き戻し処理の高速化を図る他の実施例を示す概略図である。

【図16】図15の制御回路105の処理フローを示す図である。

【図17】メモリサイズが524288バイトのフラッシュメモリの概略を示す図である。

【図18】本発明を実現するシステムの一実施例の構成を示す図である。

【図19】本発明の効果の一例を示す図である。

【図20】図14および図16の処理フローを組み合わせた処理フローを示すフローチャートである。

【図21】本発明におけるアドレスアレイのデータ保持を実現する実施例の構成図である。

【図22】図21の実施例におけるコントローラが行うアドレスアレイのデータ退避の動作フローを示す図である。

【図23】図21の実施例におけるコントローラが行うアドレスアレイおよびキャッシュメモリのデータ復帰の動作フローを示す図である。

【図24】キャッシュメモリのデータを一定時間ごとに書き戻す実施例の構成図である。

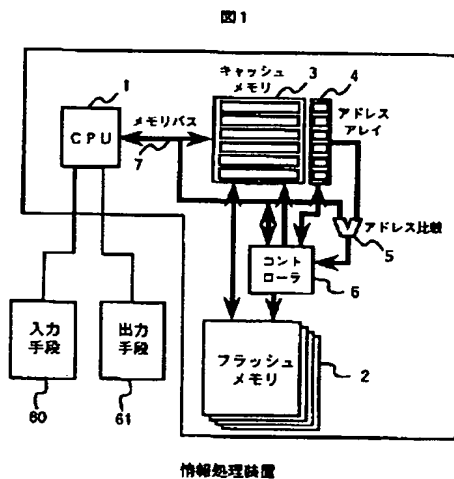
【符号の説明】

1…CPU、2…フラッシュメモリ、3…キャッシュメモリ、4…アドレスアレイ、5…アドレス比較、6…コントローラ、11…DMAコントローラ、12…メモリ制御信号発生回路、13…メモリ制御タイマ、14…データRAM、15…制御ROM、16…プロセッサ、32…フラッシュメモリアレイ、33…シリアルバッファ、38…クロック発生回路、51…フラッシュメモリ冗長領域、52…フラッシュメモリの書き込み禁止フラ

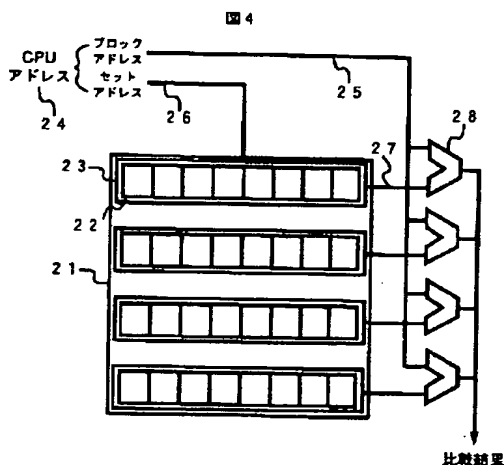
37

グ、53…キャッシュメモリの書き込み禁止フラグ、54…割込み要求信号、101…CPU、102…バス、300…コピーバック方式キャッシュメモリ、310…アドレス情報を保持するレジスタ群、311…レジスタ群310内のレジスタ、312…レジスタ群310の制御信号、a…アドレス情報を保持するアドレス部、b…ブロック消去情報を保持する消去情報部、c…更新されたか否かを示す情報を保持する更新情報部、d…更新された領域を示す情報を保持する更新領域情報部、320…データを保持するデータ群、321…レジスタ群320内のレジスタ、322…レジスタ群320の制御信号、330…アドレス比較器、331…アドレス比較器

【図1】



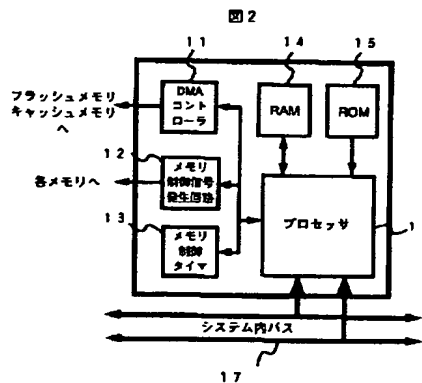
【図4】



38

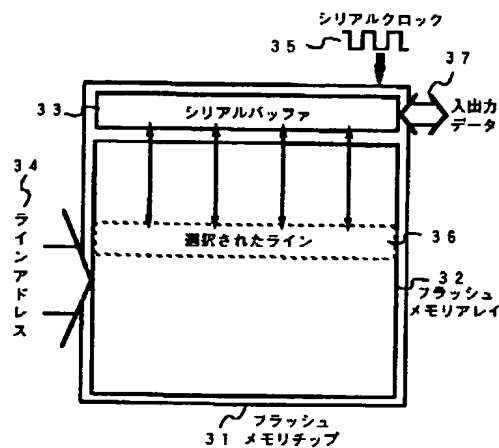
330からのキャッシュヒット判定信号、340…データを保持するレジスタ、341…レジスタ340の制御信号、350…アドレス情報を保持するレジスタ351…レジスタ350の制御信号、104…フラッシュメモリで構成される記憶装置、401…記憶装置105の制御信号、105…制御回路、222…キャッシュメモリ制御信号、A0～A8、A9～A18…アドレス信号線、I/O0～I/O7…データ信号線、201…電源スイッチ、202…電源遮断要求信号、203…アドレスアレイデータ退避領域、204…電源供給装置、205…電源遮断許可信号、211…タイマ。

【図2】

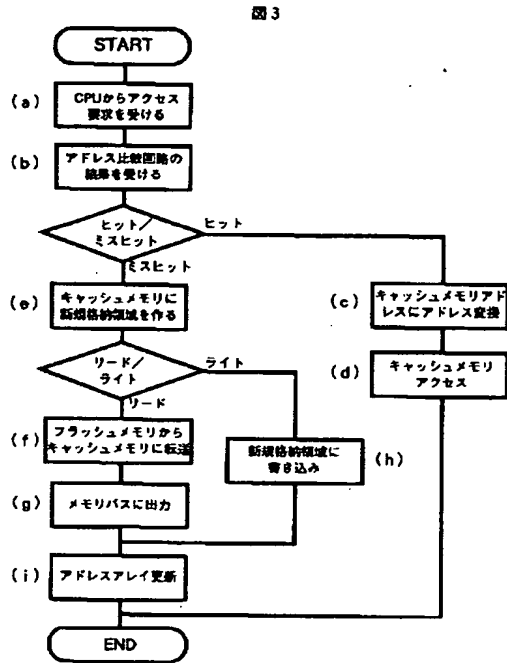


【図6】

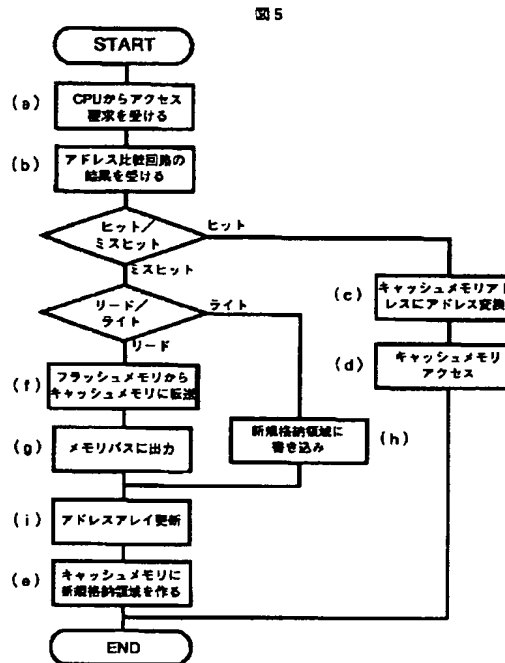
図6



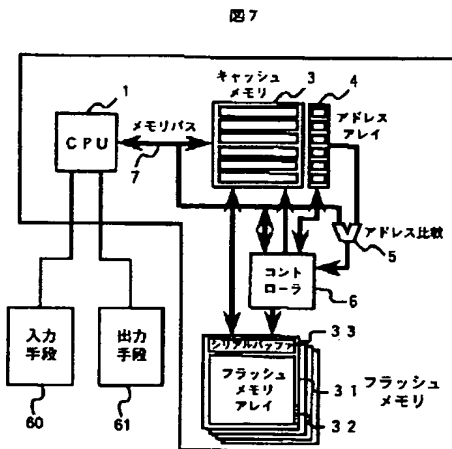
【図3】



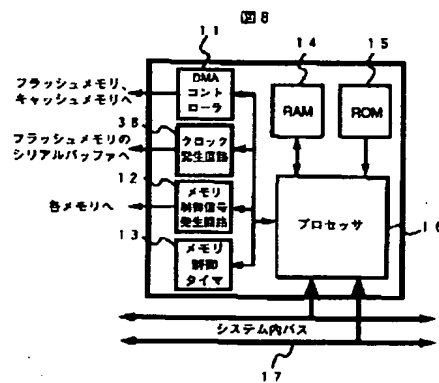
【図5】



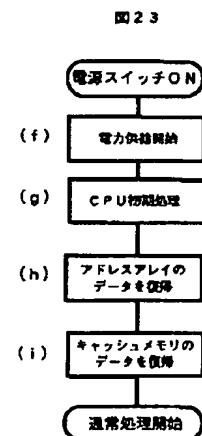
【図7】



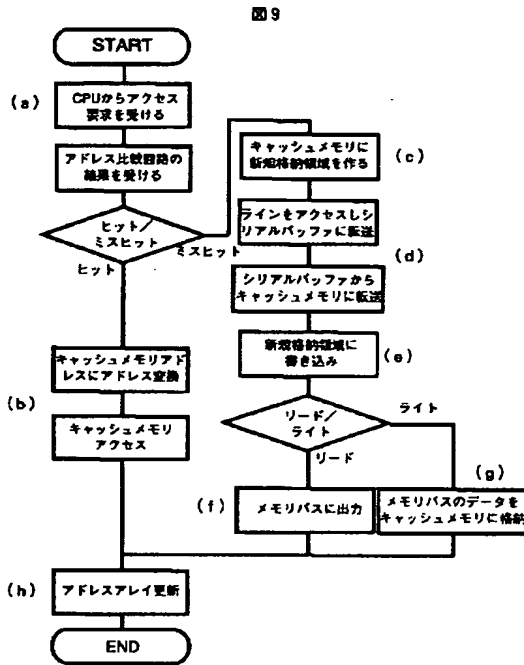
【図8】



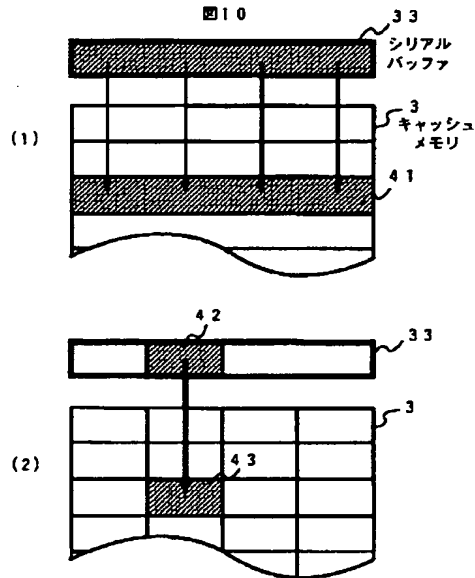
【図23】



【図9】



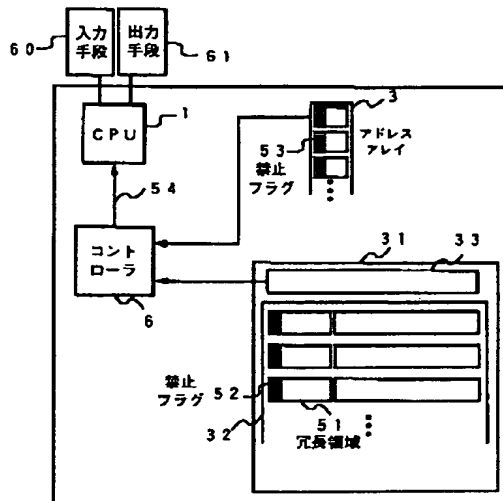
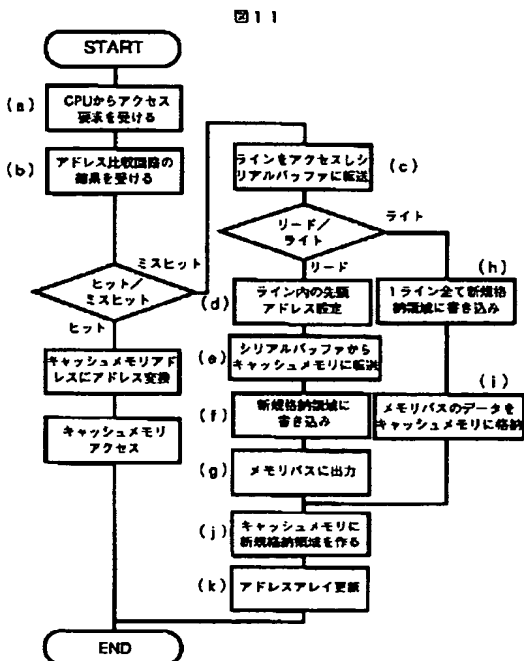
【図10】



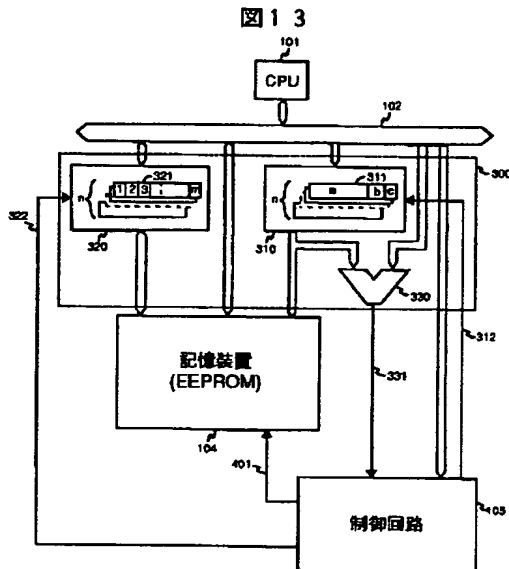
【図12】

図12

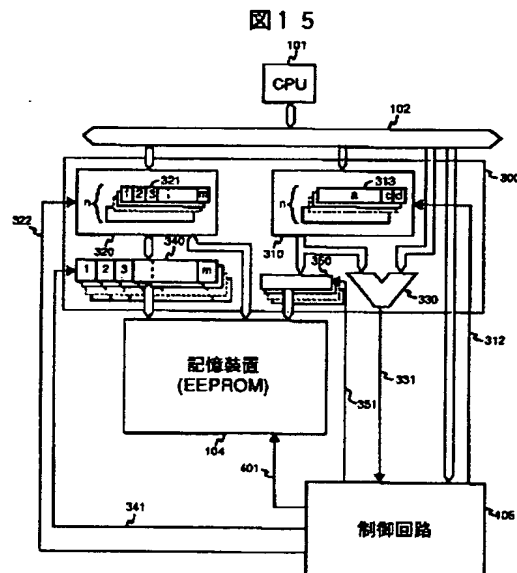
【図11】



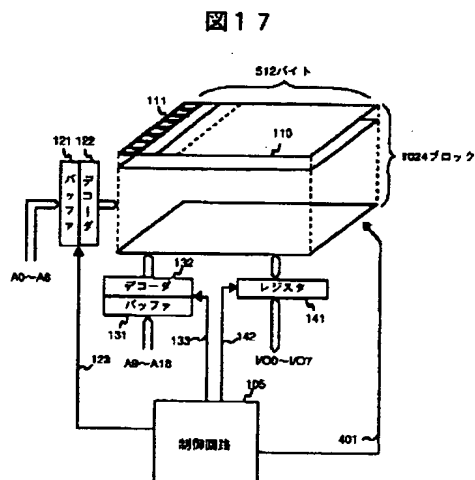
【図13】



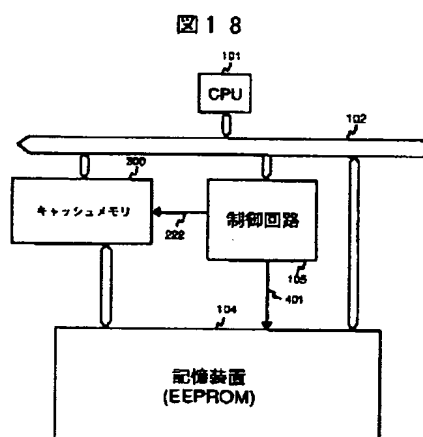
【図15】



【図17】

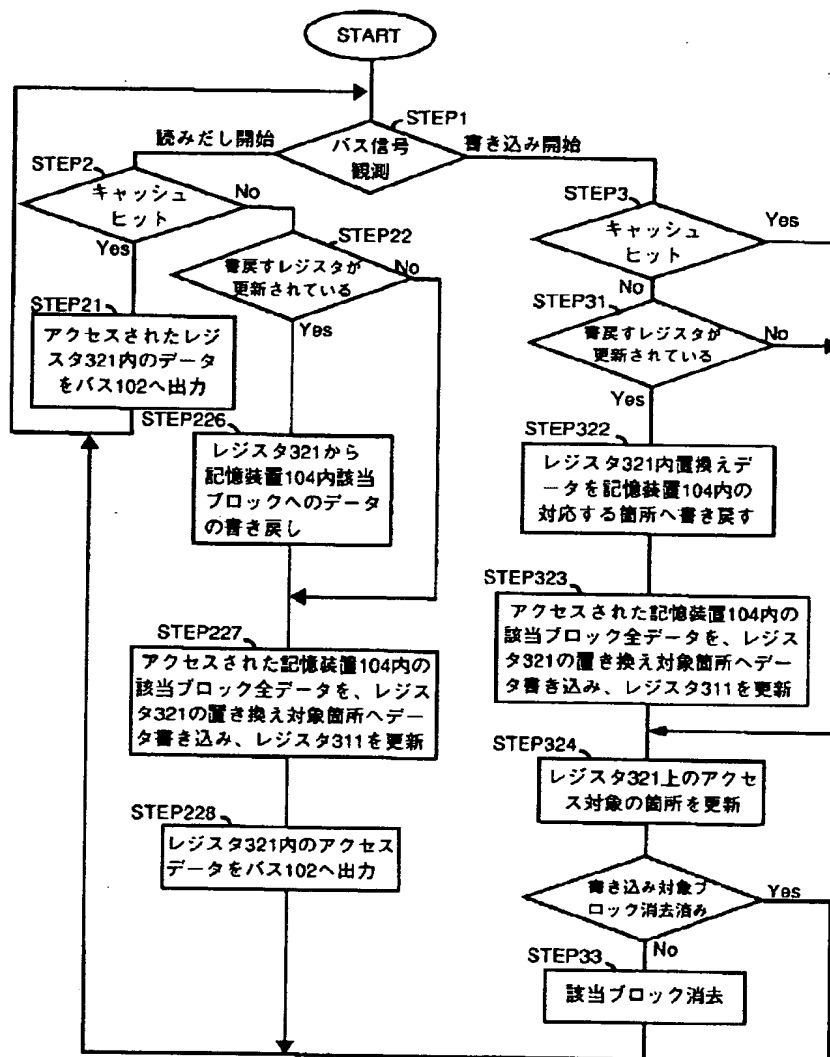


【図18】



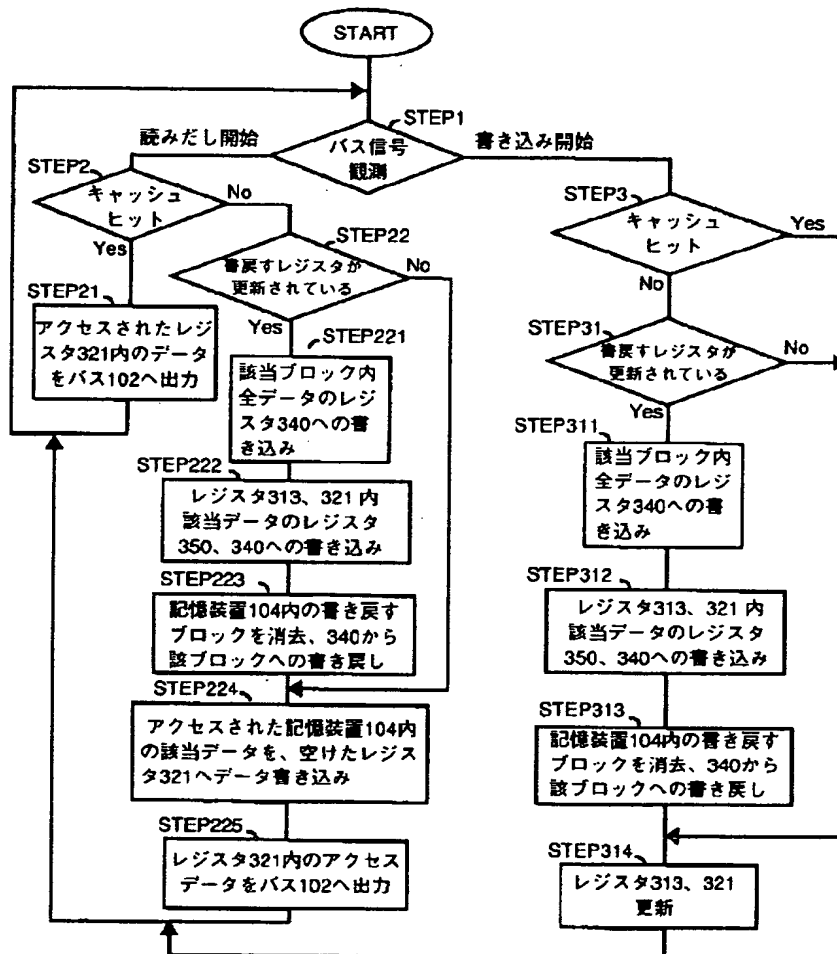
【図14】

図14



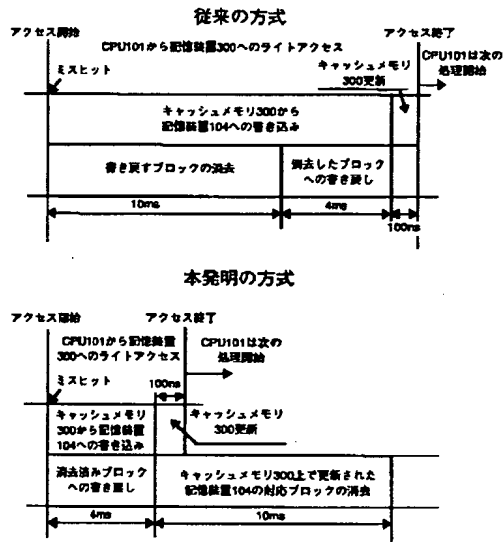
【図16】

図 16



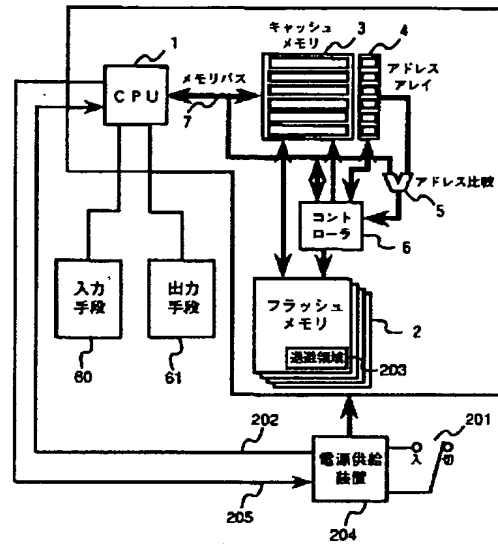
【図19】

図19



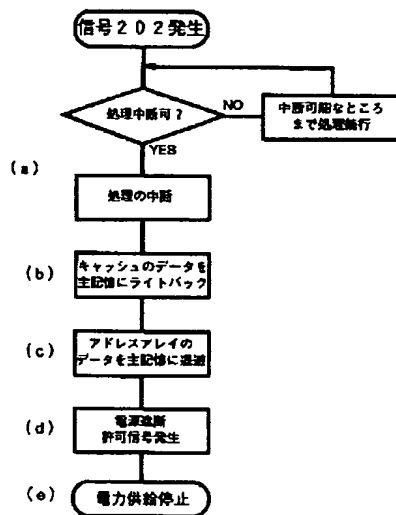
【図21】

図21



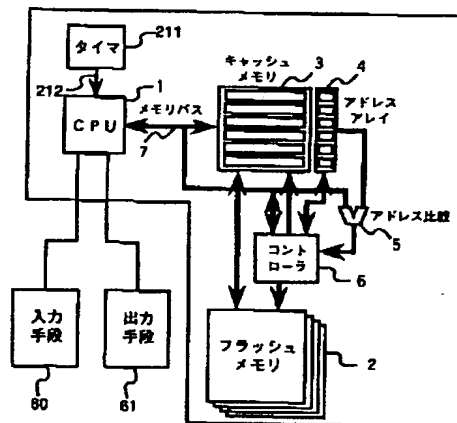
【図22】

図22



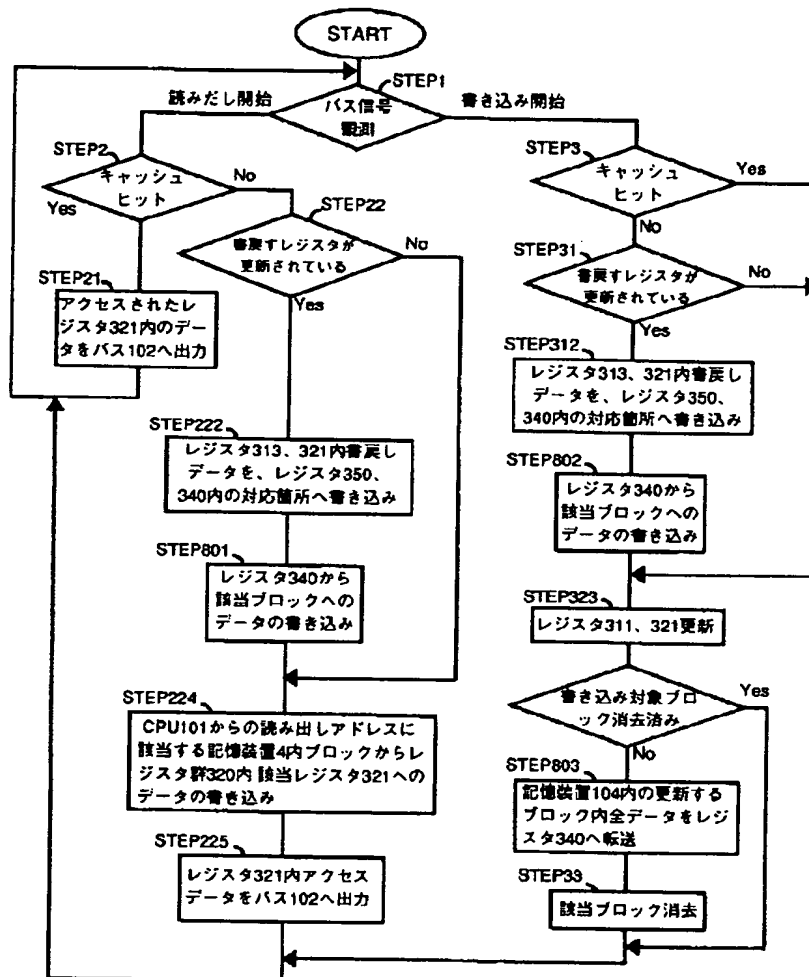
【図24】

図24



【図20】

図20



フロントページの続き

(72)発明者 服部 隆一

神奈川県川崎市麻生区王禅寺1099番地株式会社日立製作所システム開発研究所内

(72)発明者 北原 潤

神奈川県川崎市麻生区王禅寺1099番地株式会社日立製作所システム開発研究所内

(72)発明者 戸塚 隆

東京都小平市上水本町五丁目20番1号株式会社日立製作所半導体事業部内

(72)発明者 柿 健一

神奈川県海老名市下今泉810番地株式会社日立製作所オフィスシステム事業部内

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☐ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☒ **FADED TEXT OR DRAWING**

☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.